Object Transfer Profile (OTP)

Bluetooth® Test Specification

- **Issued**: 2016-12-13
- **Document Number**: OTP.TS.1.0.2
- **Group Prepared by**: Sports & Fitness WG
- **Feedback Email**: sf-main@bluetooth.org
- **Abstract**: This document defines test structures and procedures for conformance test of products implementing the Object Transfer Profile Specification.
## Revision History

<table>
<thead>
<tr>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0.5.0</td>
<td>2014-11-30</td>
<td>Initial draft based on WSP.TS.</td>
</tr>
<tr>
<td>D0.5.1</td>
<td>2015-03-01</td>
<td>Incorporated feedback from Laurence from 12/2014.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Several updates to align with current version of OTP.</td>
</tr>
<tr>
<td>D0.9.0</td>
<td>2015-03-12</td>
<td>Rolled to draft 0.9. Addressed comments from Leif.</td>
</tr>
<tr>
<td>D0.9.1</td>
<td>2015-03-15</td>
<td>Incorporated feedback from Leif.</td>
</tr>
<tr>
<td>D0.9.2</td>
<td>2015-03-17</td>
<td>Incorporated feedback from Navin and in SF WG call.</td>
</tr>
<tr>
<td>D0.9.3</td>
<td>2015-03-18</td>
<td>Incorporated significant feedback from Laurence.</td>
</tr>
<tr>
<td>D0.9.4</td>
<td>2015-03-23</td>
<td>Several major updates by Leif and Laurence.</td>
</tr>
<tr>
<td>D0.9.5</td>
<td>2015-03-23</td>
<td>Incorporated feedback from Leif and Laurence.</td>
</tr>
<tr>
<td>D0.9.6</td>
<td>2015-03-24</td>
<td>Incorporated feedback from SF WG Call. Added two new error test cases.</td>
</tr>
<tr>
<td>D0.9.7</td>
<td>2015-03-25</td>
<td>Incorporated feedback from Navin and Laurence. Replaced TCMT with content from separate document. Aligned with OTP.ICS D0.9.3.</td>
</tr>
<tr>
<td>D0.9.8</td>
<td>2015-03-26</td>
<td>Resolution of remaining SFWG comments. Preparation of a D0.9 draft for BTI review.</td>
</tr>
<tr>
<td>D0.9.9</td>
<td>2015-04-07</td>
<td>Addressed comments from BTI. Version used at Formal IOP</td>
</tr>
<tr>
<td>D1.0.0r00</td>
<td>2015-05-27</td>
<td>Addressed minor feedback from Formal IOP. Addressed deferred feedback from BTI. Fixed TCMT errors. Updated two references. Updated all references to «PSM_OTS». Added TP/OTP/CL/BV-11-I and TP/OTW/CL/BV-07-I to cover reading and writing long attribute values for the Object List Filter characteristics. Updated references to ASCII control characters. Addressed WG review comments from Leif-Alexandre Aschehoug and Charlie Lee.</td>
</tr>
<tr>
<td>D1.0.0r01</td>
<td>2015-06-22</td>
<td>Added a requirement to the Pass Verdict of TP/OTP/CL/BV-13-I [Object Discovery - Discover by Directory Listing Object] to verify that the contents of the Directory Listing Object have been correctly parsed, following discussion with the BARB.</td>
</tr>
<tr>
<td>D1.0.0r02</td>
<td>2015-09-08</td>
<td>Added an initial condition in the TP/OTP/CL/BV-13-I [Object Discovery - Discover by Directory Listing Object].</td>
</tr>
<tr>
<td>D1.0.0r03</td>
<td>2015-09-17</td>
<td>Updated TP/OTP/CL/BV-13-I following F2F discussion of feedback from the IOP performed on 7-9 September 2015.</td>
</tr>
<tr>
<td>D1.0.0r04</td>
<td>2015-10-19</td>
<td>Accepted changes from Alicia Courtney and Daniel Cowling. Resolution of BTI feedback from Alicia Courtney, Daniel Cowling and Miles Louis Smith.</td>
</tr>
<tr>
<td>1.0.0</td>
<td>2015-11-17</td>
<td>Specification adopted by BoD; prepared for publication.</td>
</tr>
</tbody>
</table>
Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Hughes</td>
<td>Intel Corporation</td>
</tr>
<tr>
<td>Laurence Richardson</td>
<td>Cambridge Silicon Radio</td>
</tr>
<tr>
<td>Leif-Alexandre Aschehoug</td>
<td>Nordic Semiconductor ASA</td>
</tr>
<tr>
<td>Navin Kochar</td>
<td>Intel Corporation</td>
</tr>
</tbody>
</table>
DISCLAIMER AND COPYRIGHT NOTICE

This disclaimer applies to all draft specifications and final specifications adopted by the Bluetooth SIG Board of Directors (both of which are hereinafter referred to herein as a Bluetooth “Specification”). Your use of this Specification in any way is subject to your compliance with all conditions of such use, and your acceptance of all disclaimers and limitations as to such use, contained in this Specification. Any user of this Specification is advised to seek appropriate legal, engineering or other professional advice regarding the use, interpretation or effect of this Specification on any matters discussed in this Specification.

Use of Bluetooth Specifications and any related intellectual property is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members, including, but not limited to, the Membership Application, the Bluetooth Patent/Copyright License Agreement and the Bluetooth Trademark License Agreement (collectively, the “Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated Bluetooth SIG and the Promoter Members (the “Early Adopters Agreements”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, “Member”) is prohibited. The use of any portion of a Bluetooth Specification may involve the use of intellectual property rights (“IPR”), including pending or issued patents, or copyrights or other rights. Bluetooth SIG has made no search for such rights and disclaims any undertaking or duty to do so. The legal rights and obligations of each Member are governed by the applicable Membership Agreements, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreements, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in (i) termination of the applicable Membership Agreements or Early Adopters Agreement and (ii) liability claims by Bluetooth SIG or any of its Members for patent, copyright and/or trademark infringement claims permitted by the applicable agreement or by applicable law.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth wireless technology (“Bluetooth Products”) may be subject to various regulatory controls under the laws and regulations applicable to products using wireless non licensed spectrum of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. To the extent not prohibited by law, in no event will Bluetooth SIG or its Members or their affiliates be liable for any damages, including without limitation, 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, arising out of or related to any furnishing, practicing, modifying, use or the performance or implementation of the contents of this Specification, even if Bluetooth SIG or its Members or their affiliates have been advised of the possibility of such damages. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS MEMBERS OR THEIR AFFILIATES RELATED TO USE OF THE SPECIFICATION.

If this Specification is an intermediate draft, it is for comment only. No products should be designed based on it except solely to verify the prototyping specification at SIG sponsored IOP events and it does not represent any commitment to release or implement any portion of the intermediate draft, which may be withdrawn, modified, or replaced at any time in the adopted Specification.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification it deems necessary or appropriate.

Copyright © 2014–2016. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. All copyrights in the Bluetooth Specifications themselves are owned by Ericsson AB, Lenovo (Singapore) Pte. Ltd., Intel Corporation, Microsoft Corporation, Apple Inc., Nokia Corporation and Toshiba Corporation. Other third-party brands and names are the property of their respective owners.
Contents

1 Scope ........................................................................................................................................... 9

2 References, Definitions, and Abbreviations .............................................................................. 10
   2.1 References .................................................................................................................................. 10
   2.2 Definitions ................................................................................................................................... 10
   2.3 Abbreviations ............................................................................................................................ 10

3 Test Suite Structure (TSS) ............................................................................................................ 11
   3.1 Overview ..................................................................................................................................... 11
   3.2 Test Strategy ............................................................................................................................... 11
   3.3 Test Groups ............................................................................................................................... 12

4 Test Cases ....................................................................................................................................... 13
   4.1 Introduction ............................................................................................................................... 13
      4.1.1 Test Case Identification Conventions ................................................................................ 13
      4.1.2 Conformance ....................................................................................................................... 13
      4.1.3 Pass/Fail Verdict Conventions ............................................................................................ 14
   4.2 Setup Preambles ....................................................................................................................... 14
      4.2.1 Set up LE Transport .......................................................................................................... 14
      4.2.2 Set up BR/EDR Transport .................................................................................................. 15
      4.2.3 Setup Object Transfer Channel ......................................................................................... 15
         4.2.3.1 Setup Object Transfer Channel over LE ..................................................................... 15
         4.2.3.2 Setup Object Transfer Channel over BR/EDR ........................................................... 15
   4.3 Service Discovery ..................................................................................................................... 15
      4.3.1 OTP/CL/OTSD/BV-01-I [Discover Object Transfer Service over LE - Primary] ................ 15
      4.3.2 OTP/CL/OTSD/BV-02-I [Discover Object Transfer Service over LE - Secondary] ........... 16
      4.3.3 OTP/CL/OTSD/BV-03-I [SDP Service Discovery] ............................................................ 17
      4.3.4 OTP/SR/OTSD/BV-04-I [Object Transfer Service not discoverable over BR/EDR] .......... 17
      4.3.5 OTP/SR/OTSD/BV-05-I [Object Transfer Service UUID in AD] ....................................... 18
   4.4 Characteristic and Descriptor Discovery .................................................................................... 18
      4.4.1 OTP/CL/OTCD/BV-01-I [Discover OTS Feature Characteristic] ......................................... 20
      4.4.2 OTP/CL/OTCD/BV-02-I [Discover Object Name Characteristic] ....................................... 20
      4.4.3 OTP/CL/OTCD/BV-03-I [Discover Object Type Characteristic] ......................................... 20
      4.4.4 OTP/CL/OTCD/BV-04-I [Discover Object Size Characteristic] ......................................... 20
      4.4.5 OTP/CL/OTCD/BV-05-I [Discover Object First-Created Characteristic] ......................... 20
      4.4.6 OTP/CL/OTCD/BV-06-I [Discover Object Last-Modified Characteristic] ....................... 20

Bluetooth SIG Proprietary and Confidential
## Characteristic Read

**4.5.1** OTP/CL/OTR/BV-01-I [Read OTS Feature Characteristic]...........................................22
**4.5.2** OTP/CL/OTR/BV-02-I [Read Object Name Characteristic]...........................................22
**4.5.3** OTP/CL/OTR/BV-03-I [Read Object Type Characteristic]...........................................22
**4.5.4** OTP/CL/OTR/BV-04-I [Read Object Size Characteristic]...........................................22
**4.5.5** OTP/CL/OTR/BV-05-I [Read Object First-Created Characteristic]...........................22
**4.5.6** OTP/CL/OTR/BV-06-I [Read Object Last-Modified Characteristic]........................22
**4.5.7** OTP/CL/OTR/BV-07-I [Read Object ID Characteristic]...........................................23
**4.5.8** OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic]...............................23
**4.5.9** OTP/CL/OTR/BV-09-I [Read Object List Filter Characteristic]...............................23
**4.5.10** OTP/CL/OTR/BV-10-I [Read Long Object Name Characteristic].............................23
**4.5.11** OTP/CL/OTR/BV-11-I [Read Long Object List Filter Characteristic]..................24
**4.5.12** OTP/CL/OTR/BI-01-I [Read OTS Feature Characteristic with Reserved Value].........25
**4.5.13** OTP/CL/OTR/BI-02-I [Read Object Properties Characteristic with Reserved Value]...26
**4.5.14** OTP/CL/OTR/BI-03-I [Read Object List Filter Characteristic with Reserved Value]....28

## Characteristic Write

**4.6.1** OTP/CL/OTW/BV-01-I [Write Object Name Characteristic]............................................30
**4.6.2** OTP/CL/OTW/BV-02-I [Write Object First-Created Characteristic]............................30
**4.6.3** OTP/CL/OTW/BV-03-I [Write Object Last-Modified Characteristic]..........................30
**4.6.4** OTP/CL/OTW/BV-04-I [Write Object Properties Characteristic].................................30
**4.6.5** OTP/CL/OTW/BV-05-I [Write Object List Filter Characteristic].................................30
**4.6.6** OTP/CL/OTW/BV-06-I [Write Long Object Name Characteristic].................................30
**4.6.7** OTP/CL/OTW/BV-07-I [Write Long Object List Filter Characteristic].......................31

## Characteristic Indications

**4.7.1** OTP/CL/OTI/BV-01-I [Configure Object Changed characteristic for Indication]........32
**4.7.2** OTP/CL/OTI/BV-02-I [Receive Object Changed Indications].....................................34
**4.7.3** OTP/CL/OTI/BV-03-I [Receive Object Changed Indications with Reserved Flags].......36
**4.7.4** OTP/CL/OTI/BV-04-I [Service Changed] ........................................................................37
4.8 Generic Object Transfer Procedures

4.8.1 OTP/CL/OTPP/BV-01-I [Object Discovery – Discover All Objects] ................................................. 38
4.8.2 OTP/CL/OTPP/BV-02-I [Object Discovery – Search for Specific Object] .......................................... 40

4.8.3 Object Discovery – Discover by Filter ............................................................................................. 41

4.8.3.1 OTP/CL/OTPP/BV-03-I [Object Discovery - Discover by Filter - Name Starts With] .............. 44
4.8.3.2 OTP/CL/OTPP/BV-04-I [Object Discovery - Discover by Filter - Name Ends With] .............. 44
4.8.3.3 OTP/CL/OTPP/BV-05-I [Object Discovery - Discover by Filter - Name Contains] ............... 44
4.8.3.4 OTP/CL/OTPP/BV-06-I [Object Discovery - Discover by Filter - Name is Exactly] ............... 44
4.8.3.5 OTP/CL/OTPP/BV-07-I [Object Discovery - Discover by Filter - Object Type] ..................... 44
4.8.3.6 OTP/CL/OTPP/BV-08-I [Object Discovery - Discover by Filter - Created between] ............. 44
4.8.3.7 OTP/CL/OTPP/BV-09-I [Object Discovery - Discover by Filter - Modified between] ............. 44
4.8.3.8 OTP/CL/OTPP/BV-10-I [Object Discovery - Discover by Filter - Current Size between] .... 44

4.8.3.9 OTP/CL/OTPP/BV-11-I [Object Discovery - Discover by Filter - Allocated Size between] ... 44
4.8.3.10 OTP/CL/OTPP/BV-12-I [Object Discovery - Discover by Filter - Marked Objects] .......... 44

4.8.5 OTP/CL/OTPP/BV-14-I [Select Object - Select by Object ID] ...................................................... 45
4.8.6 OTP/CL/OTPP/BV-15-I [Read Object - Read Object Contents] ...................................................... 46

4.8.7 Object Write .................................................................................................................................. 48

4.8.7.1 OTP/CL/OTPP/BV-16-I [Write Object - Write Object Metadata] ............................................ 48
4.8.7.2 OTP/CL/OTPP/BV-17-I [Write Object - Write Object Contents] ............................................. 49
4.8.7.3 OTP/CL/OTPP/BV-18-I [Write Object - Write Object Contents - Truncate] .......................... 51

4.8.8 Resume Read or Write Operations ................................................................................................. 52

4.8.8.1 OTP/CL/OTPP/BV-19-I [Resume Reading Object Contents] .................................................. 52
4.8.8.2 OTP/CL/OTPP/BV-20-I [Resume Writing Object Contents - Current Size Method] ............ 55
4.8.8.3 OTP/CL/OTPP/BV-21-I [Resume Writing Object Contents - Data Integrity Method] ............ 56

4.8.9 OTP/CL/OTPP/BV-22-I [Create Object] ....................................................................................... 59
4.8.10 OTP/CL/OTPP/BV-23-I [Delete Object] ......................................................................................... 60
4.8.11 OTP/CL/OTPP/BV-24-I [Execute Object] ...................................................................................... 61

4.8.12 Open Object Transfer Channel .................................................................................................. 62


4.9 Error Handling Procedures ............................................................................................................... 65
4.9.1 OACP Error Handling Procedures

4.9.1.2 OTP/CL/OAE/BI-02-I [OACP Error – Insufficient Resources] ........................................ 66
4.9.1.3 OTP/CL/OAE/BI-03-I [OACP Error – Invalid Object] ...................................................... 67
4.9.1.4 OTP/CL/OAE/BI-04-I [OACP Error – Channel Unavailable] .......................................... 68
4.9.1.5 OTP/CL/OAE/BI-05-I [OACP Error – Unsupported Type] .................................................. 69
4.9.1.6 OTP/CL/OAE/BI-06-I [OACP Error – Procedure Not Permitted] ...................................... 70
4.9.1.7 OTP/CL/OAE/BI-07-I [OACP Error – Object Locked] ....................................................... 71
4.9.1.8 OTP/CL/OAE/BI-08-I [OACP Error – Operation Failed] ................................................. 72
4.9.1.9 OTP/CL/OAE/BI-09-I [OACP Error – Procedure Timeout] .............................................. 73

4.9.2 OLCP Error Handling Procedures

4.9.2.2 OTP/CL/OLE/BI-02-I [OLCP Error – Operation Failed] .................................................. 75
4.9.2.3 OTP/CL/OLE/BI-03-I [OLCP Error – Out of Bounds] ....................................................... 75
4.9.2.4 OTP/CL/OLE/BI-04-I [OLCP Error – Too Many Objects] .................................................. 76
4.9.2.5 OTP/CL/OLE/BI-05-I [OLCP Error – No Object] ............................................................. 77
4.9.2.6 OTP/CL/OLE/BI-06-I [OLCP Error – Object ID Not Found] ............................................. 78
4.9.2.7 OTP/CL/OLE/BI-07-I [OLCP Error – Procedure Timeout] .................................................. 79

4.9.3 Application Error Handling Procedures

4.9.3.1 OTP/CL/APE/BI-01-I [Application Error – Concurrency Limit Exceeded] ...................... 80
4.9.3.2 OTP/CL/APE/BI-02-I [Application Error – Object Name Already Exists] ...................... 81

5 Test Case Mapping ............................................................................................................. 83
1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and Test Cases (TC) to test the Object Transfer Profile Specification.

The objective of this test specification is to provide a basis for interoperability tests for Bluetooth devices giving a high probability of air interface interoperability between different manufacturers’ Bluetooth devices.
2 References, Definitions, and Abbreviations

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

[1] Bluetooth Test Strategy and Terminology Overview
[2] Bluetooth Core Specification, v4.0 or later
[3] Object Transfer Profile Specification v1.0
[4] ICS Proforma for Object Transfer Profile, OTP.ICS
[7] Object Transfer Profile Implementation eXtra Information for Test, IXIT
[8] Characteristic and Descriptor descriptions and Protocol Service Multiplexer values are accessible via the Bluetooth SIG Assigned Numbers

2.2 Definitions
For the purpose of this Bluetooth document, the definitions in [1] and [2] apply.

2.3 Abbreviations
For the purpose of this Bluetooth document, the abbreviations in [1] and [2] apply.
3 Test Suite Structure (TSS)

3.1 Overview

The Object Transfer Profile requires the presence of 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”, which is used for the transfer of object contents from one device to the other. The fixed Protocol Service Multiplexer (PSM) value used for the Object Transfer protocol, «PSM_OTS», is accessible via Bluetooth Assigned Numbers [8].

![Diagram of Object Transfer Profile]

Figure 3.1: Object Transfer Test Model

3.2 Test Strategy

The test objectives are to verify functionality of the Object Transfer Profile 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 profile specification and to match these to the support of the IUT as described in the ICS Proforma.

The test equipment shall provide an implementation of the Radio Controller and the parts of the Host needed to perform the test cases defined in the Object Transfer Profile Test Specification. 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, an MMI, or another interface supported by the IUT.
The Object Transfer Profile 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.

The test suite structure is a tree with the first level representing the protocol groups. This structure is shown in Section 3.3.

Certain tests in this test specification require two independent Lower Testers to be run concurrently against the IUT.

### 3.3 Test Groups

The following test groups have been defined.

**Discovery of Services and Characteristics**

- This group tests IUT discovery of the following services and their associated characteristics:
  - Object Transfer Service and characteristics

**Features**

- This group tests the IUT implementation of Object Transfer Profile Features.

**Service Procedures**

- This group tests the operation of additional procedures defined in the service specification.

**Error Handling**

- This group tests for proper operation for handling of various error situations.
4 Test Cases

4.1 Introduction

4.1.1 Test Case Identification Conventions
Test cases shall be assigned unique identifiers per the conventions in [1]. The convention used here is `<spec abbreviation>/<IUT role>/<feat>/<xx>-<nn>-<y>.

Test group abbreviations for class, feature, function, sub-function or capability (as applicable to this test specification) are defined in Table 4.1.

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Feature Identifier &lt;feat&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTP</td>
<td>Object Transfer Profile</td>
</tr>
<tr>
<td>CL</td>
<td>Object Client Role</td>
</tr>
<tr>
<td>SR</td>
<td>Object Server Role</td>
</tr>
<tr>
<td>OTSD</td>
<td>Discovery of Services</td>
</tr>
<tr>
<td>OTCD</td>
<td>Discovery of Characteristics and Descriptors</td>
</tr>
<tr>
<td>OTR</td>
<td>Reading of Characteristics and Descriptors</td>
</tr>
<tr>
<td>OTW</td>
<td>Writing of Characteristics and Descriptors</td>
</tr>
<tr>
<td>OTI</td>
<td>Object Transfer Indications</td>
</tr>
<tr>
<td>OTF</td>
<td>OTS Features</td>
</tr>
<tr>
<td>OTPP</td>
<td>Object Transfer Procedures</td>
</tr>
<tr>
<td>OAE</td>
<td>OACP Error Handling</td>
</tr>
<tr>
<td>OLE</td>
<td>OLCP Error Handling</td>
</tr>
<tr>
<td>APE</td>
<td>Application Error Handling</td>
</tr>
</tbody>
</table>

Table 4.1: TC Feature Naming Convention for Object Transfer Profile

4.1.2 Conformance

When conformance is claimed, all capabilities indicated as mandatory for this Specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.
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 that is not excluded by the Specification, OR

• That capabilities enabled by the implementations are sustained over durations expected by the use case, OR

• That the implementation gracefully handles any quantity of data expected by the use case, OR

• That in cases where more than one valid interpretation of the Specification exist, the implementation complies with at least one interpretation and gracefully handles other interpretations OR

• That the implementation is immune to attempted security exploits.

A single execution of each of the required tests is required in order to constitute a pass verdict. However, it is noted that in order 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 Generator, the Test Case as described in the Test Specification, or with the Test System utilized, the Member is required to notify the responsible party via an errata request such that the issue may be addressed.

4.1.3 Pass/Fail Verdict Conventions

Each test case has an Expected Outcome section, which outlines all the detailed pass criteria conditions that shall be met by the IUT to merit a Pass Verdict.

The convention in this test specification 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 and in case this occurs the outcome of the test shall be the Fail Verdict.

4.2 Setup Preambles

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 Set up LE Transport

Use GATT.TS [5] Preamble [Set up ATT Bearer over LE].
4.2.2 Set up BR/EDR Transport
Use GATT.TS [5] Preamble [Set up ATT Bearer over BR/EDR].

4.2.3 Setup Object Transfer Channel
The Protocol Service Multiplexer (PSM) value «PSM_OTS» used in the following procedures represents the fixed PSM value for the Object Transfer protocol, as defined in [8].

4.2.3.1 Setup Object Transfer Channel over LE
Preamble procedure:
1. The Upper Tester induces the IUT to open an Object Transfer Channel.
2. The IUT sends a LE_Credit_Based_Connection_Request with the PSM parameter set to «PSM_OTS» and other parameters set to valid values.
3. After receiving the LE_Credit_Based_Connection_Request, the Lower Tester sends a LE_Credit_Based_Connection_Rsp.

4.2.3.2 Setup Object Transfer Channel over BR/EDR
Preamble procedure:
1. The Upper Tester induces the IUT to open an Object Transfer Channel.
2. The IUT sends an L2CAP_ConnectReq with the PSM parameter set to «PSM_OTS» and other parameters set to valid values for Enhanced Retransmission Mode (ERTM).
3. After receiving the L2CAP_ConnectReq, the Lower Tester sends an L2CAP_ConnectRsp.

4.3 Service Discovery
The procedures defined in this test group verify IUT’s ability to discover the services exposed by an Object Server (Lower Tester).

4.3.1 OTP/CL/OTSD/BV-01-I [Discover Object Transfer Service over LE - Primary]
• Test Purpose
  Verify that the Object Transfer Service can be discovered by the Object Client IUT when using an LE Transport when the service is a Primary service.
• Reference
  [3] 4.2
• Initial Condition
  Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1.
The Lower Tester exposes one instantiation of the Object Transfer Service [6].

• Test Procedure

1. The IUT executes the Discover All Primary Services sub-procedure by executing the test procedure included in GATT test case GATT/SR/GAD/BV-01-C [Discover All Primary Services - from Server] in [5]. Alternatively, the IUT executes the Discover Primary Services by Service UUID sub-procedure by executing the test procedure included in GATT test case GATT/SR/GAD/BV-02-C [Discover Primary Services by Service UUID - from server] in [5] with the Service UUID set to «Object Transfer Service».

• Expected Outcome

Pass verdict
Verify that the attribute handle range discovered by the IUT contains the starting handle and the ending handle of the instantiation of the Object Transfer Service.

4.3.2 OTP/CL/OTSD/BV-02-I [Discover Object Transfer Service over LE - Secondary]

• Test Purpose

Verify that the Object Transfer Service can be discovered by the Object Client IUT when using an LE Transport when the service is a Secondary (included) service.

• Reference

[3] 4.2

• Initial Condition

Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1.

The Lower Tester exposes one instantiation of the Object Transfer Service [6].

• Test Procedure

1. The IUT discovers the service as a secondary service by executing the procedure described in GATT test case GATT/CL/GAD/BV-03-C [Find Included Services – by client] in [5], until the Include declaration containing the service UUID «Object Transfer Service» in the response is found.

• Expected Outcome

Pass verdict
Verify that the attribute handle range discovered by the IUT, contains the starting handle and the ending handle of the instantiation of the Object Transfer Service.
4.3.3  OTP/CL/OTSD/BV-03-I [SDP Service Discovery]

- Test Purpose
  Verify that the Object Client IUT can discover the SDP record for the Object Transfer Service of the Lower Tester when using the BR/EDR transport.

- Reference
  [3] 4.2

- Initial Condition
  Establish a BR/EDR ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.2.

- Test Procedure
  1. The IUT establishes an SDP connection to the Lower Tester.
  2. The IUT sends SDP requests to retrieve attributes of the SDP record for the Object Transfer Service.
  3. Monitor the SDP requests sent by the IUT to retrieve attributes of SDP records from the Lower Tester.

- Expected Outcome
  Pass verdict
  The SDP record for the Object Transfer Service is retrieved.

4.3.4  OTP/SR/OTSD/BV-04-I [Object Transfer Service not discoverable over BR/EDR]

- Test Purpose
  Verify that the Object Transfer Service on a BR/EDR/LE (i.e. dual mode) Object Server IUT that only supports the service over LE cannot be discovered by an Object Client when using a BR/EDR based ATT Bearer.

- Reference
  [3] 4.2

- Initial Condition
  The IUT includes one instantiation of the Object Transfer Service [6].

- Test Procedure
  1. Establish a BR/EDR ATT Bearer connection between the Lower Tester and IUT as described in Section 4.2.2.
2. The Lower Tester uses the SDP Service Discovery procedure to attempt to discover the services supported by the IUT over BR/EDR.

- Expected Outcome

  **Pass verdict**
  No SDP record for the Object Transfer Service is discovered over BR/EDR.

### 4.3.5 OTP/SR/OTSD/BV-05-I [Object Transfer Service UUID in AD]

- **Test Purpose**

  Verify that the Object Transfer Service UUID is included in AD (Advertising Data) from the Object Server IUT when using the LE Transport.

- **Reference**

  [3] 3.1.1

- **Initial Condition**

  The IUT is powered on in GAP Discoverable Mode.
  The IUT is induced to generate Advertising Packets.

- **Test Procedure**

  The Lower Tester listens for Advertising Packets from the IUT.

- **Expected Outcome**

  **Pass verdict**
  At least one received Advertising Packet contains the defined Service UUID for «Object Transfer Service».

### 4.4 Characteristic and Descriptor Discovery

- **Test Purpose**

  This test group contains test cases to verify that the characteristics and descriptors claimed in the ICS [4] can be discovered by the Object Client. The verification is performed one at a time, as enumerated in the test cases in Table 4.2, using this generic test procedure.

- **Reference**

  [3] 4.3

- **Initial Condition**
Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics. This instantiation also contains two “future” characteristics. Each “future” characteristic has a 16-bit UUID randomly selected from unassigned UUIDs at the time of the test.

The IUT has discovered the Object Transfer Service and has saved the handle range for an instantiation of the Object Transfer Service. That instantiation contains an instantiation of the claimed OTS characteristic. This was done by previously using the GATT method in OTP/CL/OTSD/BV-01-I [Discover Object Transfer Service over LE - Primary] or OTP/CL/OTSD/BV-02-I [Discover Object Transfer Service over LE - Secondary] for the LE transport, or using the SDP method in OTP/CL/OTSD/BV-03-I [SDP Service Discovery] for the BR/EDR transport.

• Test Procedure

1. The Upper Tester issues a command to the IUT to Discover the OTS characteristic represented in the test case.

2. The IUT executes either of the procedures included in GATT.TS [5]: GATT/CL/GAD/BV-04-C [Discover All Characteristic of a Service], with the specified handle range for the instantiation of the Object Transfer Service, or GATT/CL/GAD/BV-05-C [Discover Characteristic by UUID], with the specified handle range for the instantiation of the Object Transfer Service and UUID set to the supported characteristic.

3. If the properties of the characteristic support indication, then the Upper Tester issues a command to the IUT to Discover All Characteristic Descriptors using the handle range of the characteristic. The IUT executes one pass of the procedure included in GATT.TS [5]: GATT/CL/GAD/BV-06-C [Discover all Characteristic Descriptors] using the specified handle range, with the server database defined in Initial Condition.

• Expected Outcome

The following pass verdict applies to the test cases listed in Table 4.2.

Pass verdict

For all characteristics other than the Object List Filter characteristic, verify that one attribute handle/value pair is received by the IUT containing the UUID associated with the supported characteristic.

For the Object List Filter characteristic, verify that three attribute handle/value pairs are received by the IUT containing the UUIDs associated with the Object List Filter characteristic.
It is acceptable for additional handle/value pairs for other supported characteristics to be received (and ignored for the purposes of the test case).

For characteristics that support indications (i.e., OLCP, OACP and Object Changed), verify that one attribute handle/value pair is received by the IUT containing the UUID «Client Characteristic Descriptor».

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Additional Test Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.1 OTP/CL/OTCD/BV-01-I [Discover OTS Feature Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.2 OTP/CL/OTCD/BV-02-I [Discover Object Name Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.3 OTP/CL/OTCD/BV-03-I [Discover Object Type Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.4 OTP/CL/OTCD/BV-04-I [Discover Object Size Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.5 OTP/CL/OTCD/BV-05-I [Discover Object First-Created Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.6 OTP/CL/OTCD/BV-06-I [Discover Object Last-Modified Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.7 OTP/CL/OTCD/BV-07-I [Discover Object ID Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.8 OTP/CL/OTCD/BV-08-I [Discover Object Properties Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.9 OTP/CL/OTCD/BV-09-I [Discover Object Action Control Point (OACP) Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.10 OTP/CL/OTCD/BV-10-I [Discover Object List Control Point (OLCP) Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.11 OTP/CL/OTCD/BV-11-I [Discover Object List Filter Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.4.12 OTP/CL/OTCD/BV-12-I [Discover Object Changed Characteristic]</td>
<td>None.</td>
</tr>
</tbody>
</table>

*Table 4.2: Characteristic Discovery Test Cases*
4.5 Characteristic Read

This test group contains test cases to verify that the characteristics that are readable and claimed in the ICS [4] can be read by the Object Client. The verification is performed one at a time, as enumerated in the test cases in Table 4.3, using this generic test procedure.

- Reference

[3] 4.4

- Initial Condition

All characteristics of OTS supported by the IUT are specified in the ICS [4].

Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

Each characteristic contains valid data. The test values used for UTF-8 based characteristics are non-zero length strings and do not include any ASCII control character.

The IUT has previously executed the procedure included in Section 4.4, so it has the handle/value pairs for all characteristics of the Object Transfer Service exposed by the Lower Tester that are supported by the IUT.

The length of each characteristic used in this test case is such that its value can be read in its entirety in a GATT Read transaction when the default ATT_MTU size is used.

- Test Procedure

1. The Upper Tester issues a command to the IUT to read the supported characteristic represented from the Lower Tester.


3. After receipt of the \textit{ATT\_Read\_Response} (0x0B) from the Lower Tester, the IUT sends the value of the supported characteristic to the Upper Tester.
• Expected Outcome

The following pass verdict applies to the test cases listed in Table 4.3.

Pass verdict

The IUT sends a correctly formatted ATT_Read_Request (0x0A) to the Lower Tester, containing the handle specified by the Upper Tester for each supported characteristic.

The IUT receives the response from the Lower Tester and sends the ReadResponse containing the correct characteristic value to the Upper Tester.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Additional Test Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5.1 OTP/CL/OTR/BV-01-I [Read OTS Feature Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.2 OTP/CL/OTR/BV-02-I [Read Object Name Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.3 OTP/CL/OTR/BV-03-I [Read Object Type Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.4 OTP/CL/OTR/BV-04-I [Read Object Size Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.5 OTP/CL/OTR/BV-05-I [Read Object First-Created Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.6 OTP/CL/OTR/BV-06-I [Read Object Last-Modified Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>Test Case</td>
<td>Additional Test Requirements</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>4.5.7 OTP/CL/OTR/BV-07-I [Read Object ID Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.8 OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic]</td>
<td>None.</td>
</tr>
<tr>
<td>4.5.9 OTP/CL/OTR/BV-09-I [Read Object List Filter Characteristic]</td>
<td>Lower Tester contains a valid combination of values for the three Object List Filter characteristics.</td>
</tr>
</tbody>
</table>

Table 4.3: Characteristic Read Test Cases

4.5.10 OTP/CL/OTR/BV-10-I [Read Long Object Name Characteristic]

- **Test Purpose**
  
  Verify that an Object Client IUT can use the GATT Read Long sub-procedure successfully to read the Object Name characteristic that contains a value in UTF-8 format that exceeds the capacity of default ATT_MTU size.

- **Reference**
  
  [3] 4.1

- **Initial Condition**
  
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to set up the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT has previously executed the procedure included in Section 4.4, so it has the handle/value pairs for all characteristics of the Object Transfer Service exposed by the Lower Tester that are supported by the IUT.

  The Lower Tester does not permit an ATT_MTU size larger than the default ATT_MTU size to be negotiated.

  The length of the Object Name characteristic used in this test case is sufficiently long that its value cannot be read in its entirety in a GATT Read transaction when the default ATT_MTU size is used and therefore requires the GATT Read Long sub-procedure to be used.

  The UTF-8 string used in the Object Name value is a non-zero length string and does not include any ASCII control character.

- **Test Procedure**
1. The Upper Tester issues a command to the IUT to read the Object Name characteristic.

2. The IUT executes the procedure included in GATT.TS [5] GATT/CL/GAR/BV-04-C
   [Read Long Characteristic Value - by client].

   • Expected Outcome

   Pass verdict
   The IUT reports the Object Name characteristic value correctly, in its entirety.

4.5.11 OTP/CL/OTR/BV-11-I [Read Long Object List Filter Characteristic]

   • Test Purpose

   Verify that an Object Client IUT can use the GATT Read Long sub-procedure successfully to
   read an Object List Filter characteristic that contains a value with a length that exceeds the
   capacity of the default ATT_MTU size.

   • Reference

   [3] 4.5.2.2

   • Initial Condition

   Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if
   using a BR/EDR transport, to set up the transport and initiate connection to an Object
   Server.

   The Lower Tester includes one instantiation of the Object Transfer Service [6] including all
   defined characteristics.

   The IUT has previously executed the procedure included in Section 4.4, so it has the
   handle/value pairs for all characteristics of the Object Transfer Service exposed by the
   Lower Tester that are supported by the IUT.

   The Lower Tester does not permit an ATT_MTU size larger than the default ATT_MTU size
   to be negotiated.

   The Lower Tester contains a valid combination of values for the three Object List Filter
   characteristics.

   The length of at least one of the Object List Filter characteristics used in this test case is
   sufficiently long that its value cannot be read in its entirety in a GATT Read transaction
   when the default ATT_MTU size is used and therefore requires the GATT Read Long sub-
   procedure to be used. Therefore, the test value used is more than (ATT_MTU-1) octets in
   length. This is achieved by including a sufficiently long string value in the parameter field.

   The filter condition used in the Object List Filter characteristic that contains the long attribute
   value shall be ‘Name Starts With’, ‘Name Ends With’, ‘Name Contains’ or ‘Name is Exactly’.
The UTF-8 string used in the parameter of the Object List Filter characteristic value is a non-zero length string and does not include any ASCII control character.

- **Test Procedure**

  1. The Upper Tester issues a command to the IUT to read the Object List Filter characteristic(s).

  2. For the long attribute value(s), the IUT executes the procedure included in GATT.TS [5] GATT/CL/GAR/BV-04-C [Read Long Characteristic Value - by client] to read the characteristic. Otherwise, the IUT executes the procedure included in GATT.TS [5] GATT/CL/GAR/BV-01-C [Read Characteristic Value – by client].

  3. The IUT repeats step #2 until all three Object List Filter characteristics have been read.

- **Expected Outcome**

  **Pass verdict**

  Even though three instances of the Object List Filter characteristic are exposed, it is sufficient for the purposes of this test case if the IUT reads the value of one instance, provided that at least one long attribute value is read during the test.

  In step #2, the IUT detects when the attribute value to be read is long and executes the GATT Read Long sub-procedure. The IUT reads the entire value of the Object List Filter characteristic successfully.

### 4.5.12 OTP/CL/OTR/BI-01-I [Read OTS Feature Characteristic with Reserved Value]

- **Test Purpose**

  Verify that the Object Client IUT is tolerant to the presence of RFU values for the OTS Feature characteristic.

- **Reference**

  [3] 4.6

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The IUT knows the handle of the OTS Feature characteristic contained in the Lower Tester.

  For the purposes of this test case, a valid OTS Feature characteristic value is used, but also with at least one RFU bit from each field set to 1.
• Test Procedure

1. Send a command from Upper Tester to request the IUT to read the OTS Feature Characteristic from the Lower Tester.

2. After receipt of the expected result by the Lower Tester from the IUT, send an \textit{ATT\_Read\_Response} (0x0B) from the Lower Tester to the IUT containing a value in which at least two RFU bits are non-zero.

- **Expected Outcome**

  \textbf{Pass} verdict

  The IUT sends a correctly formatted \textit{ATT\_Read\_Request} (0x0A) to the Lower Tester, containing the handle specified by the Upper Tester.

  The IUT receives the response from the Lower Tester, ignores the RFU values and continues to function in accordance with the requirements of the profile.

4.5.13 **OTP/CL/OTR/BI-02-I [Read Object Properties Characteristic with Reserved Value]**

• Test Purpose

  Verify that the Object Client IUT is tolerant to the presence of RFU values for the Object Properties characteristic.
• Reference

[3] 4.6

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The IUT knows the handle of the Object Properties characteristic contained in the Lower Tester.

For the purposes of this test case, a valid Object Properties characteristic value is used, but also with at least one RFU bit set to 1.

• Test Procedure

1. Send a command from Upper Tester to request the IUT to read the Object Properties Characteristic from the Lower Tester.

2. After receipt of the expected result by the Lower Tester from the IUT, send an ATT_Read_Response (0x0B) from the Lower Tester to the IUT containing a value in which at least one RFU bit is non-zero.

• Expected Outcome

Pass verdict

The IUT sends a correctly formatted ATT_Read_Request (0x0A) to the Lower Tester, containing the handle specified by the Upper Tester.

The IUT receives the response from the Lower Tester, ignores the RFU value(s) and continues to function in accordance with the requirements of the profile.
4.5.14 OTP/CL/OTR/BI-03-I [Read Object List Filter Characteristic with Reserved Value]

- **Test Purpose**
  
  Verify that the Object Client IUT is tolerant to the presence of RFU values for at least one Object List Filter characteristic.

- **Reference**
  
  [3] 4.6

- **Initial Condition**
  
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The IUT knows the handle of all Object List Filter characteristics contained in the Lower Tester.

  For the purposes of this test case, valid Object List Filter characteristic values are used, but with at least one from the pool of RFU values, the characteristic(s) with the RFU value, also includes a Parameter that is 1 to 20 octets in length.

- **Test Procedure**
  
  1. Send a command from Upper Tester to request the IUT to read all Object List Filter characteristics from the Lower Tester.

  2. After receipt of each expected result by the Lower Tester from the IUT, send an ATT_Read_Response (0x0B) from the Lower Tester to the IUT containing an RFU value in at least one of the Object List Filter characteristics.

- **Expected Outcome**
Pass verdict

The IUT sends a correctly formatted \textit{ATT\_Read\_Request} (0x0A) to the Lower Tester for each of the three characteristics, containing the handle specified by the Upper Tester.

The IUT receives the response from the Lower Tester, ignores the RFU value(s) and corresponding parameters(s) and continues to function in accordance with the requirements of the profile.

4.6 Characteristic Write

This test group contains test cases to verify that the characteristics that are writable and claimed in the ICS [4] can be written to by the Object Client. The verification is performed one at a time, as enumerated in the test cases in Table 4.4, using this generic test procedure.

- Reference

[3] 4.5.5.1

- Initial Condition

All characteristics of OTS for which writing is supported by the IUT are specified in the ICS [4].

Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics. Each characteristic that supports writing includes the object properties metadata with the ‘Write’ bit set to 1.

The IUT has previously executed the procedure included in Section 4.4, so it has the handle/value pairs for all characteristics of the Object Transfer Service exposed by the Lower Tester that are supported by the IUT.

The IUT may need to perform a re-discovery of characteristics prior to performing this test group in case the database has changed since previous tests were performed.

The length of the test value to be written is such that it can be written in its entirety in a GATT Write transaction, using the default minimum ATT\_MTU size.

- Test Procedure

1. The Upper Tester issues a command to the IUT to read (one at a time) the supported characteristics from the Lower Tester.

2. The Upper Tester issues a command to the IUT to write a new value to the supported characteristics.
3. For each characteristic of the Object Transfer Service supported by the IUT, the IUT executes the procedure included in GATT.TS [5] GATT/CL/GAW/BV-03-C [Write Characteristic Value – by client].

- Expected Outcome

The following pass verdict applies to the test cases listed in Table 4.4.

**Pass verdict**

The IUT writes a correctly formatted characteristic value to the Lower Tester.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Additional Test Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6.1 OTP/CL/OTW/BV-01-I [Write Object Name Characteristic]</td>
<td>IUT writes a non-zero length value that does not include any ASCII control character.</td>
</tr>
<tr>
<td>4.6.2 OTP/CL/OTW/BV-02-I [Write Object First-Created Characteristic]</td>
<td>IUT writes a valid date and time value.</td>
</tr>
<tr>
<td>4.6.3 OTP/CL/OTW/BV-03-I [Write Object Last-Modified Characteristic]</td>
<td>IUT writes a valid date and time value.</td>
</tr>
<tr>
<td>4.6.5 OTP/CL/OTW/BV-05-I [Write Object List Filter Characteristic]</td>
<td>IUT writes a valid combination of values to the three Object List Filter characteristics.</td>
</tr>
</tbody>
</table>

**Table 4.4: Characteristic Write Test Cases**

**4.6.6 OTP/CL/OTW/BV-06-I [Write Long Object Name Characteristic]**

- **Test Purpose**

Verify that an Object Client IUT can use the GATT Write Long sub-procedure successfully to write to an Object Name characteristic that contains a value in UTF-8 format when the length of the value to be written exceeds the capacity of default ATT_MTU size.

- **Reference**

[3] 4.5.5.1

- **Initial Condition**

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.
The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics. The object properties metadata of the Object Name characteristic has the ‘Write’ bit set to 1.

The IUT has previously executed the procedure included in Section 4.4, so it has the handle/value pairs for all characteristics of the Object Transfer Service exposed by the Lower Tester that are supported by the IUT.

The Lower Tester does not permit an ATT_MTU size larger than the default ATT_MTU size to be negotiated.

The length of the Object Name characteristic used in this test case is such that its value cannot be written in its entirety in a GATT Write transaction when the default ATT_MTU size is used and therefore requires the GATT Write Long procedure to be used.

• Test Procedure

1. The Upper Tester issues a command to the IUT to write a new value to the Object Name characteristic.


• Expected Outcome

Pass verdict

The IUT writes the correct Object Name characteristic value to the Lower Tester in its entirety by using the GATT Write Long sub-procedure.

The value written is a non-zero length string and does not include any ASCII control character.

4.6.7 OTP/CL/OTW/BV-07-I [Write Long Object List Filter Characteristic]

• Test Purpose

Verify that an Object Client IUT can use the GATT Write Long sub-procedure successfully to write to an Object List Filter characteristic when the length of the value to be written exceeds the capacity of default ATT_MTU size.

• Reference

[3] 4.5.2.2

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to set up the transport and initiate connection to an Object Server.
The Lower Tester does not permit an ATT_MTU size larger than the default ATT_MTU size to be negotiated.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics. The Lower Tester exposes three Object List Filter characteristics.

The IUT has previously executed the procedure included in Section 4.4, so it has the handle/value pairs for all characteristics of the Object Transfer Service exposed by the Lower Tester that are supported by the IUT.

The filter condition to be written to the Object List Filter characteristic shall be ‘Name Starts With’, ‘Name Ends With’, ‘Name Contains’ or ‘Name is Exactly’. The UTF-8 string used in the parameter of the Object List Filter characteristic value is a non-zero length string and does not include any ASCII control character.

• The length of the value to be written to the Object List Filter characteristic is sufficiently long that it cannot be written in its entirety in a GATT Write transaction when the default ATT_MTU size is used and therefore requires the GATT Write Long sub-procedure to be used. Therefore, the test value used is more than (ATT_MTU-3) octets in length. This is achieved by including a sufficiently long string value in the parameter field.

• Test Procedure
  1. The Upper Tester issues a command to the IUT to write a new value to at least one of the Object List Filter characteristics.

• Expected Outcome

  Pass verdict
  The IUT writes the correct Object List Filter characteristic value to the Lower Tester in its entirety by using the GATT Write Long sub-procedure.
  The value written is a non-zero length string and does not include any ASCII control character.

  Note: Although three instances of the Object List Filter characteristic are exposed, verifying that a long attribute value can be written to at least one instance of the characteristic is sufficient for the purposes of this test case.

4.7 Characteristic Indications

The procedures defined in this test group verify an Object Client IUT can configure an Object Server for indications and receive indications.

4.7.1 OTP/CL/OTI/BV-01-I [Configure Object Changed characteristic for Indication]

• Test Purpose
Verify that the Object Client IUT can configure an Object Server (Lower Tester) to indicate the Object Changed characteristic.

- **Reference**
  
  [3] 4.4.3

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handle of the Object Changed characteristic.

  The IUT has discovered the Client Characteristic Configuration Descriptor for the Object Changed characteristic contained in the Lower Tester.

- **Test Procedure**

  1. The Upper Tester sends a command to the IUT to configure the Object Server to send Object Changed characteristics.

  2. IUT sends a correctly formatted ATT_Write_Request (0x12) to the Lower Tester, with the handle set to that of the Client Characteristic Configuration descriptor for the Object Changed characteristic with the value set to “Indication”.

  3. The Lower Tester sends an ATT_Write_Response (0x13) to the IUT.

- **Expected Outcome**

  Pass verdict
In step #2, the IUT configures the Client Characteristic Configuration descriptor of the Object Changed characteristic for indications.

### 4.7.2 OTP/CL/OTI/BV-02-I [Receive Object Changed Indications]

- **Test Purpose**
  
  Verify that the Object Client IUT can receive indications of the Object Changed characteristic, including variants.

- **Reference**
  
  [3] 4.4.3

- **Initial Condition**
  
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT has discovered the Client Characteristic Configuration Descriptor of the Object Changed characteristic contained in the Lower Tester.

  The IUT is put into a state in which it is prepared to receive Object Changed indications, by executing the procedure described in OTP/CL/OTI/BV-01-I [Configure Object Changed characteristic for Indication] in Section 4.7.1 or by other means.

  The IUT knows the handle of the Object Changed characteristic.

- **Test Procedure**

  1. For each Test Pattern shown in Table 4.5, the Lower Tester sends an ATT_Handle_Value_Indication containing an Object Changed characteristic value to the IUT. The test values for the Flags field and the corresponding pass criteria are shown for each test pattern. The value of the Object ID field is a 48-bit integer randomly selected from pool of non-RFU values.

     For each case, the IUT receives the indication and sends a Handle Value Confirmation.

<pre><code> | Test Pattern | Flags Field Value (bit7..bit0) | Pass Criteria |
 |--------------|-------------------------------|---------------|
 | 1            | 0000 0110                     | Source of Change: Server |
 |              |                               | Change occurred to the object contents: True |
 |              |                               | Change occurred to the object metadata: True |
 |              |                               | Object Creation: False |
 |              |                               | Object Deletion: False |
</code></pre>
<table>
<thead>
<tr>
<th>Test Pattern</th>
<th>Flags Field Value (bit7..bit0)</th>
<th>Pass Criteria</th>
</tr>
</thead>
</table>
| 2            | 0000 0111                      | Source of Change: Client  
Change occurred to the object contents: True  
Change occurred to the object metadata: True  
Object Creation: False  
Object Deletion: False |
| 3            | 0000 1000                      | Source of Change: Server  
Change occurred to the object contents: False  
Change occurred to the object metadata: False  
Object Creation: True  
Object Deletion: False |
| 4            | 0000 1001                      | Source of Change: Client  
Change occurred to the object contents: False  
Change occurred to the object metadata: False  
Object Creation: True  
Object Deletion: False |
| 5            | 0001 0000                      | Source of Change: Server  
Change occurred to the object contents: False  
Change occurred to the object metadata: False  
Object Creation: False  
Object Deletion: True |
| 6            | 0001 0001                      | Source of Change: Client  
Change occurred to the object contents: False  
Change occurred to the object metadata: False  
Object Creation: False  
Object Deletion: True |

*Table 4.5: Object Changed Test Patterns*
• Expected Outcome

Pass verdict
IUT is able to correctly parse the received Object Changed values according to the pass criteria in Table 4.5 above.
The reported Flags and Object ID field values match the ones sent by the Lower Tester.
In all cases, the IUT continues to process commands and to receive data normally.

4.7.3  OTP/CL/OTI/BV-03-I [Receive Object Changed Indications with Reserved Flags]

• Test Purpose
Verify that the Object Client IUT can receive indications of the Object Changed characteristic from an Object Server including reserved flags.

• Reference
[3] 4.4.3

• Initial Condition
Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.
The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.
The IUT is put into a state in which it is prepared to receive Object Changed indications, by executing the procedure described in OTP/CL/OTI/BV-01-I [Configure Object Changed characteristic for Indication] in Section 4.7.1 or by other means.

The IUT knows the handle of the Object Changed characteristic.

- **Test Procedure**

  The Lower Tester sends an `ATT_Handle_Value_Indication` to the IUT containing a valid Object Changed characteristic value and at least one RFU flag set to 1.

  ![Diagram](process_diagram)

  - **Expected Outcome**

    **Pass verdict**

    IUT reports the received Object Changed characteristic value to the Upper Tester. The reported Object Changed value matches the value sent by the Lower Tester.

    The IUT ignores the RFU value(s) and continues to function in accordance with the requirements of the profile.

4.7.4  **OTP/CL/OTI/BV-04-I [Service Changed]**

- **Test Purpose**

  Verify that the Object Client IUT re-performs service and characteristic discovery following receipt of a Serviced Changed indication from an Object Server with which it has a trusted relationship and that it refreshes cached values.
• Reference

[3] 4.6

• Initial Condition

The IUT and the Lower Tester have previously bonded and have a trusted relationship.

The IUT has previously read the OTS Feature characteristic as described in OTP/CL/OTR/BV-01-I [Read OTS Feature Characteristic] in Section 4.5.1.

The Lower Tester has the Service Changed characteristic.

The IUT has configured the Service Changed characteristic for indications as defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication].

No connection is established between the IUT and Lower Tester.

• Test Procedure

1. Run the preamble procedure to enable the Object Client to initiate connection to an Object Server included in the Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport.

2. The Lower Tester indicates the Service Changed characteristic. The characteristic value contains the beginning and ending attribute handles for the service definition including the Object Transfer Service.

3. The IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

4. The IUT re-performs service and characteristic discovery.

• Expected Outcome

Pass verdict

The IUT successfully re-discovers all of the supported characteristics and characteristic descriptors of the Object Transfer Service.

The IUT re-reads the OTS Feature characteristic by sending a correctly formatted ATT_Read_Request (0x0A) to the Lower Tester, containing the refreshed handle value for the OTS Feature characteristic.

The IUT receives the response from the Lower Tester and sends the ReadResponse containing the correct OTS Feature value to the Upper Tester.

4.8 Generic Object Transfer Procedures

4.8.1 OTP/CL/OTPP/BV-01-I [Object Discovery – Discover All Objects]

• Test Purpose

Verify that the Object Client IUT can perform the Discover All Objects sub-procedure.
• Reference

[3] 4.5.2.1

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handle of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

The Lower Tester contains at least 3 valid objects.

The Current Object is other than the first object in the list of objects.

The IUT may initially set the List Sort Order so that the object discovery process described in the following steps occurs in an order as specified by the IUT.

• Test Procedure

1. The IUT writes a value of 0x00 ("No Filter") to all three instances of the Object List Filter characteristic.

2. The IUT writes the OLCP First Op Code (0x01) to the OLCP without a Parameter Value to select the first object.

3. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

4. The IUT reads at least one of the Object Metadata characteristics such as the Object Name or Object ID or Object Type.

5. The IUT writes the OLCP Next Op Code (0x04) to the OLCP without a Parameter Value to select the next object.

6. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x04) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

The IUT repeats steps 4, 5 and 6 until receiving an Error Response with the Result Code set to ‘Out of Bounds’ (0x05).
• Expected Outcome

Pass verdict
The IUT writes a correctly formatted First op code to the OLCP.

The IUT reads at least one Object Metadata characteristic each time a new object is selected.

The IUT writes a correctly formatted Next op code to the OLCP until all objects have been discovered.

After receiving each indication of the OLCP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

4.8.2 OTP/CL/OTPP/BV-02-I [Object Discovery – Search for Specific Object]

• Test Purpose
Verify that the Object Client IUT can perform the Search for Specific Object sub-procedure.

• Reference
[3] 4.5.2.3

• Initial Condition
Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

The Lower Tester contains at least 3 valid objects.

The object which the Upper Tester will specify as the target object to be found is not the first object in the list of objects.

Each object contains valid metadata. Where used, the object name metadata is set to a non-zero length value and does not include any ASCII control character.

The IUT may initially set the List Sort Order so that the object discovery process described in the following steps occurs in an order as specified by the IUT.

• Test Procedure
1. The Upper Tester instructs the IUT to search for a specific object, providing the IUT with the name of one of the objects that is available from the Lower Tester.

2. The IUT writes a value of 0x00 (“No Filter”) to all three instances of the Object List Filter characteristic.

3. The IUT writes the OLCP First Op Code (0x01) to the OLCP without a Parameter Value to select the first object.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

5. The IUT reads at least the Object Name characteristic.

6. The IUT writes the OLCP Next Op Code (0x04) to the OLCP without a Parameter Value to select the next object.

7. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x04) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

8. The IUT repeats steps 5, 6 and 7 until the object specified by the Upper Tester in step #1 has been discovered.

   The IUT informs the Upper Tester that an object has been discovered as specified by the search criteria.

• Expected Outcome

   Pass verdict

   The IUT writes “No Filter” to all three instances of the Object List Filter characteristic.

   The IUT writes a correctly formatted First op code to the OLCP.

   The IUT reads at least one Object Metadata characteristic each time a new object is selected.

   The IUT writes a correctly formatted Next op code to the OLCP until the specific object has been discovered. At the end of the procedure, the object that was specified by the Upper Tester has become the Current Object.

   After receiving each indication of the OLCP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

4.8.3 Object Discovery – Discover by Filter

• Test Purpose
This test group contains test cases to verify compliant operation when using the Discover by Filter sub-procedure for different filter conditions that are claimed in the ICS [4]. The verification is performed one at a time, as enumerated in the test cases in Table 4.6, using this generic test procedure.

- Reference

[3] 4.5.2.2

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

The Lower Tester contains at least 3 valid objects.

Each object contains valid metadata. Where used, the object name metadata is set to a non-zero length value and does not include any ASCII control character.

The IUT may initially set the List Sort Order so that the object discovery process described in the following steps occurs in an order as specified by the IUT.

- Test Procedure

1. The Upper Tester instructs the IUT to search for all objects whose metadata satisfy specified filter conditions.

2. The IUT writes a valid value (Filter Value and Parameter) to at least one of the Object List Filter characteristics according to the test in Table 4.6.

3. The IUT writes the OLCP First Op Code (0x01) to the OLCP without a Parameter Value to select the first object.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

5. The IUT reads at least the Object Name characteristic.

6. The IUT writes the OLCP Next Op Code (0x04) to the OLCP without a Parameter Value to select the next object.
7. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x04) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

8. The IUT repeats steps 5, 6 and 7 until receiving an Error Response with the Result Code set to ‘Out of Bounds’ (0x05).

   The IUT informs the Upper Tester that an object has been discovered as specified by the search criteria.

   • Expected Outcome

   Pass verdict

   The IUT writes a valid value (Filter Value and Parameter) to at least one of the Object List Filter characteristics and the combination of values established in the three Object List Filter characteristics matches the filter requirements specified by the Upper Tester.

   The IUT writes a correctly formatted First op code to the OLCP.

   The IUT reads at least one Object Metadata characteristic each time a new object is selected.

   The IUT writes a correctly formatted Next op code to the OLCP until all the objects in the filtered list of objects have been discovered.

   After receiving each indication of the OLCP, the IUT sends an `ATT_Handle_Value_Confirmation` to the Lower Tester.
<table>
<thead>
<tr>
<th>Test Case</th>
<th>Additional Test Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8.3.1</td>
<td>Set the Filter Value to 0x01 (Name Starts With)</td>
</tr>
<tr>
<td>4.8.3.2</td>
<td>Set the Filter Value to 0x02 (Name Ends With)</td>
</tr>
<tr>
<td>4.8.3.3</td>
<td>Set the Filter Value to 0x03 (Name Contains)</td>
</tr>
<tr>
<td>4.8.3.4</td>
<td>Set the Filter Value to 0x04 (Name is Exactly)</td>
</tr>
<tr>
<td>4.8.3.5</td>
<td>Set the Filter Value to 0x05 (Object Type)</td>
</tr>
<tr>
<td>4.8.3.6</td>
<td>Set the Filter Value to 0x06 (Created between)</td>
</tr>
<tr>
<td>4.8.3.7</td>
<td>Set the Filter Value to 0x07 (Modified between)</td>
</tr>
<tr>
<td>4.8.3.8</td>
<td>Set the Filter Value to 0x08 (Current Size between)</td>
</tr>
<tr>
<td>4.8.3.9</td>
<td>Set the Filter Value to 0x09 (Allocated Size between)</td>
</tr>
<tr>
<td>4.8.3.10</td>
<td>Set the Filter Value to 0x0A (Marked Objects) Lower Tester contains at least one object with the 'Mark' bit of the 'object properties' metadata set to 1.</td>
</tr>
</tbody>
</table>

Table 4.6: Object Discovery by Filter Test Cases

4.8.4 OTP/CL/OTPP/BV-13-I [Object Discovery - Discover by Directory Listing Object]

- Test Purpose

  Verify that the Object Client IUT can perform the Discover by Directory Listing Object sub-procedure.
• Reference

[3] 4.5.2.4

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP and OLCP for indications.

The Lower Tester contains the Directory Listing Object.

The Lower Tester contains at least 3 valid objects other than the Directory Listing Object.

At least the first Object Record contained within the Directory Listing Object has one of the RFU bits of its Flags field set to one and also has more than one Extension octet present.

• Test Procedure

1. The Upper Tester induces the IUT to read the Directory Listing Object.

2. The IUT selects the Directory Listing Object using the Select by Object ID sub-procedure (see OTP/CL/OTPP/BV-14-I [Select Object - Select by Object ID] in Section 4.8.5) using the Object ID of the Directory Listing Object (i.e., 0x000000000000) as the parameter.

3. The IUT reads the Current Object using the Read Object Content sub-procedure (see OTP/CL/OTPP/BV-15-I [Read Object - Read Object Contents] in Section 4.8.6).

• Expected Outcome

Pass verdict

The IUT writes a correctly formatted Go To Op Code to the OLCP with the Directory Listing Object ID (i.e., 0x000000000000) as the parameter.

After receiving indication of the OLCP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

The IUT correctly receives the Directory Listing Object via the Object Transfer Channel and correctly parses the contents of the Directory Listing Object in accordance with the service.

4.8.5 OTP/CL/OTPP/BV-14-I [Select Object - Select by Object ID]

• Test Purpose
Verify that the Object Client IUT can perform the Select by Object ID sub-procedure.

- **Reference**
  - [3] 4.5.3.2

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP or the OLCP for indications.

  The Lower Tester contains at least one valid object with object ID set to Object ID 1 (for example 0x000000000002).

- **Test Procedure**

  1. The IUT writes the OLCP Go To Op Code (0x05) to the OLCP with Object ID 1 as the parameter (for example 0x000000000002).

  2. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

  3. The IUT sends an ATThandle_Value_Confirmation.

- **Expected Outcome**

  **Pass verdict**

  The IUT writes a correctly formatted Go To Op Code with the Object ID 1 as the parameter.

  After indication of the OLCP, the IUT sends an ATThandle_Value_Confirmation to the Lower Tester.

`4.8.6 OTP/CL/OTPP/BV-15-I [Read Object - Read Object Contents]`

- **Test Purpose**

  Verify that the Object Client IUT can perform the Read Object Contents sub-procedure.
• Reference

• [3] 4.5.4.2

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object with valid content that supports the Read property, and this object is selected.

• Test Procedure

1. The Upper Tester induces the IUT to read the part or the whole of the contents of the Current Object.

2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4).

3. The IUT writes the OACP Read Op Code (0x05) to the OACP with the Offset parameter set to a value less than the Current Size field read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Current Size field read in step #2.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

5. The IUT sends an ATT_Handle_Value_Confirmation.

6. The Lower Tester sends data consisting of the number of octets specified in the Length parameter (see step #3) starting from the offset specified in the Offset Parameter (see step #3) to the IUT via the Object Transfer Channel.

• Expected Outcome

Pass verdict
The IUT writes a correctly formatted Read Op Code (0x05) with the Offset parameter less than the Current Size field read in step #2 and a Length parameter such that the sum of the Length and the Offset parameters is less than or equal to the Current Size field read in step #2.

The Offset and Length values written by the IUT are consistent with the request made by the Upper Tester.

After indication of the OACP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

The specified object contents are correctly received via the Object Transfer Channel by the IUT.

4.8.7 Object Write

4.8.7.1 OTP/CL/OTPP/BV-16-I [Write Object - Write Object Metadata]

- Test Purpose

Verify the aspects of the Write Object Metadata sub-procedure that are not exercised by performing the individual test cases in the ‘Writing of Characteristics and Descriptors’ (OTW) test group.

Verify that the Object Client IUT can perform the Write Object Metadata sub-procedure when it attempts to write a long attribute value and the request is not supported by the Server.

- Reference

[3] 4.5.5.1

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including the Object Name characteristic. The object properties metadata of the Object Name characteristic has the ‘Write’ bit set to 1.

The IUT has previously discovered the handle/value pairs for all characteristics used in the following test procedure.

The Lower Tester does not permit an ATT_MTU size larger than the default ATT_MTU size to be negotiated.

The length of the Object Name string value initially used in this test case is such that its value cannot be written in its entirety in a single GATT Write transaction when the default
ATT_MTU size is used and therefore requires the GATT Write Long procedure to be used. The string does not include any ASCII control character.

• Test Procedure

1. The Upper Tester issues a command to the IUT to write a new value to the Object Name characteristic. The value requested in this step induces the IUT to attempt the GATT Write Long procedure.


3. The Lower Tester rejects the `ATT_Prepare_Write_Request` by responding with the ATT Error Response `ATT_Request_Not_Supported`.

4. If requested by the IUT, the Upper Tester provides a shorter name that is short enough to be written without requiring the GATT Write Long procedure to be used.

5. The IUT writes the shorter object name to the Object Name characteristic by executing the procedure included in GATT.TS [5] GATT/CL/GAW/BV-03-C [Write Characteristic Value – by client].

• Expected Outcome

Pass verdict
After receiving the error response in step #3, the IUT returns to a stable state and can process commands normally.

In step #4, it is acceptable for the IUT either to generate a shorter name automatically (e.g., by truncating the longer name) or to request input from the Upper Tester.

In step #5, the IUT successfully writes the shorter object name to the Object Name characteristic.

4.8.7.2  OTP/CL/OTPP/BV-17-I [Write Object - Write Object Contents]

• Test Purpose
Verify that the Object Client IUT can perform the Write Object Contents sub-procedure without truncation.
• Reference

[3] 4.5.5.2

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object that supports the Write and Patch properties, and this object is selected.

• Test Procedure

1. The Upper Tester induces the IUT to write to the Current Object.
2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4) and the Object Properties characteristic (see OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic] in Section 4.5.8).
3. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value less than or equal to the Current Size read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Allocated Size read in step #2 and the Mode parameter value set to not truncate (bit 1 is set to 0).
4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.
5. The IUT sends an ATT_Handle_Value_Confirmation.
6. The IUT sends data to the Lower Tester via the Object Transfer Channel.
7. The Lower Tester waits for the object transfer operation to complete.

• Expected Outcome

Pass verdict
The IUT writes a correctly formatted Write Op Code (0x06) with the Offset parameter less than or equal to the Current Size read in step #2 and a length parameter such that the sum of the length and the offset is less than or equal to the Allocated Size read in step #2.

The Offset and Length values written by the IUT are consistent with the request made by the Upper Tester.

After indication of the OACP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

The specified object contents are received by the Lower Tester via the Object Transfer Channel.

The IUT sends the expected number of octets to the Lower Tester.

4.8.7.3 OTP/CL/OTPP/BV-18-I [Write Object - Write Object Contents - Truncate]

- Test Purpose

Verify that the Object Client IUT can perform the Write Object Contents sub-procedure with truncation.

- Reference

[3] 4.5.5.2

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object that supports the Write, Patch and Truncate properties, and this object is selected.

- Test Procedure

1. The Upper Tester induces the IUT to truncate the Current Object.

2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4).
3. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value less than the Current Size read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than the Current Size read in step #2 and the Mode parameter value set to truncate (bit 1 is set to 1).

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for 'Success' (0x01) without a Response Parameter.

5. The IUT sends an ATT_Handle_Value_Confirmation.

6. The IUT sends data to the Lower Tester via the Object Transfer Channel.

7. The Lower Tester waits for the object transfer operation to complete.

- Expected Outcome

Pass verdict

The IUT writes correctly formatted Write Op Code (0x06) with the Offset parameter value less than the Current Size value read in step #2 and a length parameter such that the sum of the Length and Offset parameter values is less than the Current Size value read in step #2 and the Mode parameter value set to truncate (bit 1 is set to 1).

The Offset and Length values written by the IUT are consistent with the request made by the Upper Tester.

After indication of the OACP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

The specified object contents are received by the Lower Tester via the Object Transfer Channel.

The IUT sends the expected number of octets to the Lower Tester.

4.8.8 Resume Read or Write Operations

4.8.8.1 OTP/CL/OTPP/BV-19-I [Resume Reading Object Contents]

- Test Purpose

Verify that the Object Client IUT can perform the Resume Reading Object Contents sub-procedure.
• **Reference**

[3] 4.5.6.1

• **Initial Condition**

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object with valid content that supports the Read property, and this object is selected.

• **Test Procedure**

1. The Upper Tester induces the IUT to read the part or the whole of the contents of the Current Object.

2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4) and the Object Properties characteristic (see OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic] in Section 4.5.8).

3. The IUT writes the OACP Read Op Code (0x05) to the OACP with the Offset parameter set to a value less than the Current Size read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Current Size read in step #2.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for 'Success' (0x01) without a Response Parameter.

5. The IUT sends an **ATT_Handle_Value_Confirmation**.

6. The Lower Tester sends data consisting of a number of octets less than specified in the Length parameter (see step #3) starting from the offset specified in the Offset parameter (see step #3) to the IUT via the Object Transfer Channel.

7. The Lower Tester disconnects before the object transfer is able to complete.
8. The Upper Tester induces the IUT to resume the interrupted transfer, unless resumption is automatic.

9. A new connection is established.

10. The IUT selects the object (using one of the Select Object Sub-procedures).

11. The IUT writes the OACP Calculate Checksum Op Code (0x03) with the Offset parameter set to the same value as in step #3 and a Length parameter less than the length value used in step #3 (this value should be consistent with the number of octets received by the IUT before the disconnection).

12. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x03) followed by the Result Code for ‘Success’ (0x01) and the Checksum parameter set to a value equal to the checksum calculated on the data for the length received in step #11.

13. The IUT writes the OACP Read Op Code (0x05) to the OACP with the Offset parameter set to a value equal to the sum of the length and offset used in step #11 and the Length Parameter set to a value such that the sum of Length and Offset parameter is equal to the sum of the length and offset parameters used in step #3.

14. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

15. The IUT sends an ATT_Handle_Value_Confirmation

16. The Lower Tester sends data consisting of the number of octets specified in the Length parameter used in step #13 starting from the offset specified in the Offset parameter used in step #13 to the IUT via the Object Transfer Channel.

• Expected Outcome

Pass verdict

In step #11, the IUT sets the Offset parameter to the same value as in step #3 and the Length parameter to a value less than or equal to the number of bytes sent by the Lower Tester prior to the disconnection.

In step #13, the IUT sets the Offset parameter to a value equal to the sum of the Offset and Length parameters used in step #11 and a Length parameter such that the sum of this length and of the offset is equal to the sum of the Length and Offset parameters used in step #3.

After step #16, the IUT has received all of the object contents that were specified by the Offset and Length parameters in step #3 and has correctly reassembled these contents in one object.
4.8.8.2 OTP/CL/OTPP/BV-20-I [Resume Writing Object Contents - Current Size Method]

- **Test Purpose**
  
  Verify that the Object Client IUT can perform the Resume Writing Object Contents sub-procedure.

- **Reference**

  [3] 4.5.6.2

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

  Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

  The Lower Tester contains at least one valid object with valid content that supports the Write and Patch properties, and this object is selected. A small or empty object is used so that its size increases as required by the test procedure.

- **Test Procedure**

  1. The Upper Tester induces the IUT to write to the Current Object, without truncation.

  2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4) and the Object Properties characteristic (see OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic] in Section 4.5.8).

  3. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value less than or equal to the Current Size read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Allocated Size read in step #2 and the Mode parameter value set to not truncate (bit 1 is set to 0).

  4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.
5. The IUT sends an ATT_Handle_Value_Confirmation.

6. The IUT starts sending data to the IUT via the Object Transfer Channel.

7. Once the current size of the object has started increasing, the Lower Tester disconnects before the object transfer is able to complete.

8. The Upper Tester induces the IUT to resume the interrupted transfer, unless resumption is automatic.

9. A new connection is established.

10. The IUT selects the object (using one of the Select Object Sub-procedures).

11. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4).

12. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value equal to the Current Size read in step #11 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is equal to the sum of the Length and Offset parameters used in step #3 and the Mode parameter value set to not truncate (bit 1 is set to 0).

13. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

14. The IUT sends an ATT_Handle_Value_Confirmation.

15. The Lower Tester receives data consisting of the number of octets specified in the Length parameter used in step #12 starting from the offset specified in the Offset parameter used in step #12 from the IUT via the Object Transfer Channel.

• Expected Outcome

Pass verdict

In step #11, the IUT reads the Object Size characteristic

In step #12, the IUT sets the Offset parameter to a value equal to the Current Size read in step #11 and a Length parameter such that the sum of this length and of the offset is equal to the sum of the Length and Offset parameters used in step #3.

In step #15, the number of octets received by the Lower Tester is equal to the length used in step #12.

4.8.8.3 OTP/CL/OTPP/BV-21-I [Resume Writing Object Contents - Data Integrity Method]

• Test Purpose

Verify that the Object Client IUT can perform the Resume Writing Object Contents sub-procedure.
• Reference

[3] 4.5.6.2

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object with valid content that supports the Write and Patch properties, and this object is selected.

• Test Procedure

1. The Upper Tester induces the IUT to write to the Current Object, without truncation.

2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4) and the Object Properties characteristic (see OTP/CL/OTR/BV-08-I [Read Object Properties Characteristic] in Section 4.5.8).

3. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value less than or equal to the Current Size read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Allocated Size read in step #2 and the Mode parameter value set to not truncate (bit 1 is set to 0).

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for 'Success' (0x01) without a Response Parameter.

5. The IUT sends an ATT_Handle_Value_Confirmation.

6. The IUT starts sending data to the IUT via the Object Transfer Channel.

7. The Lower Tester disconnects before the object transfer is able to complete.

8. The Upper Tester induces the IUT to resume the interrupted transfer, unless resumption is automatic.
9. A new connection is established.

10. The IUT selects the object (using one of the Select Object Sub-procedures).

11. The IUT reads the Object Size characteristic (see [Read Object Size Characteristic] in Section 4.5.4).

12. The IUT writes the OACP Calculate Checksum Op Code (0x03) with the Offset parameter set to the same value as in step #3 and a Length parameter less than or equal to the Current Size read in step #11.

13. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x03) followed by the Result Code for ‘Success’ (0x01) and the Checksum parameter set to a value not matching the checksum for the data covered by the used length and offset.

14. The IUT writes the OACP Calculate Checksum Op Code (0x03) again with updated parameter values.

15. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x03) followed by the Result Code for ‘Success’ (0x01) and the Checksum parameter set to a value matching the checksum for the data covered by the used length and offset.

16. The IUT may repeat steps 14 and 15 multiple times while it is checking the data integrity of the object contents that have already been written to the Lower Tester, according to its own method.

17. The IUT writes the OACP Write Op Code (0x06) to the OACP with the Offset parameter set to a value greater than the Offset parameter value used in step #3 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is equal to the sum of the length and offset parameters used in step #3 and the Mode parameter value set to not truncate (bit 1 is set to 0).

18. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x06) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

19. The IUT sends an `ATT_Handle_Value_Confirmation`.

20. The Lower Tester receives data consisting of the number of octets specified in the Length parameter used in step #17 starting from the offset specified in the Offset parameter used in step #17 from the IUT via the Object Transfer Channel.

- Expected Outcome

  **Pass verdict**
In step #11, the IUT reads the Object Size characteristic.

In step #12, the IUT sends a correctly formatted Calculate Checksum Op Code to the OACP with the Offset parameter to the same value as in step #3 and a Length parameter less than or equal to the Current Size read in step #11.

In step #14, the IUT sends a correctly formatted Calculate Checksum Op Code to the OACP with valid parameter values.

In step #17, the IUT sets the Offset parameter to a value greater than the Offset parameter value used in step #3 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is equal to the sum of the length and offset parameters used in step #3 and the Mode parameter value set to not truncate (bit 1 is set to 0).

In step #20, the number of octets received by the Lower Tester is equal to the length used in step #17.

**4.8.9 OTP/CL/OTPP/BV-22-I [Create Object]**

- **Test Purpose**
  
  Verify that the Object Client IUT can perform the Create Object procedure.

- **Reference**
  
  [3] 4.5.7

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- **Test Procedure**

  1. The Upper Tester induces the IUT to create a new object.

  2. The IUT writes the OACP Create Op Code (0x01) to the OACP with the Type parameter set to valid object type (e.g., the UUID “Firmware”) and the Size Parameter set to a valid value (e.g., 0x00000400, representing an allocated object size of 1024 octets).

  3. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code.
Op Code (0x60), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

4. The IUT sends an ATT_Handle_Value_Confirmation.

5. The IUT writes the Object Name characteristic with a non-zero length UTF-8 string (see OTP/CL/OTW/BV-01-I [Write Object Name Characteristic] in Section 4.6.1).

6. The Lower Tester sends a Write Response to acknowledge the write to the Object Name.

- Expected Outcome

Pass verdict

The IUT writes correctly formatted Create Op Code to the OACP.

After receiving indication of the OACP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

The IUT writes a non-zero length string to the Object Name characteristic. In addition, the IUT may also write values to other writable Object Metadata characteristics.

4.8.10 OTP/CL/OTPP/BV-23-I [Delete Object]

- Test Purpose

Verify that the Object Client IUT can perform the Delete Object procedure.

- Reference

[3] 4.5.8

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

The Lower Tester contains at least one valid object with valid content that supports the Delete property, and this object is selected.
• Test Procedure

1. The Upper Tester induces the IUT to Delete the current object.
2. The IUT writes the OACP Delete Op Code (0x02) to the OACP without any parameter.
3. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x02) followed by the Result Code for 'Success' (0x01) without a Response Parameter.
4. The IUT sends an \textit{ATT\_Handle\_Value\_Confirmation}.

• Expected Outcome

\textbf{Pass verdict}

The IUT writes correctly formatted Delete Op Code to the OACP.
After receiving indication of the OACP, the IUT sends an \textit{ATT\_Handle\_Value\_Confirmation} to the Lower Tester.

\textbf{4.8.11 OTP/CL/OTPP/BV-24-I [Execute Object]}

• Test Purpose

Verify that the Object Client IUT can perform the Execute Object procedure.

• Reference

[3] 4.5.9

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

The Lower Tester contains at least one valid object with valid content that supports the Execute property, and this object is selected.

• Test Procedure

1. The Upper Tester induces the IUT to Execute the current object.
2. The IUT writes the OACP Execute Op Code (0x04) to the OACP without any parameter.

3. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x04) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

4. The IUT sends an _ATT_Handle_Value_Confirmation_.

   • Expected Outcome

   **Pass verdict**
   
   The IUT writes correctly formatted Execute Op Code to the OACP.

   After receiving indication of the OACP, the IUT sends an _ATT_Handle_Value_Confirmation_ to the Lower Tester.

---

**4.8.12 Open Object Transfer Channel**

**4.8.12.1 OTP/CL/OTPP/BV-25-I [Open Object Transfer Channel - LE transport]**

• **Test Purpose**

Verify that the Object Client IUT can perform the Open Object Transfer Channel procedure over LE transport.

• **Reference**

[3] 4.5.10

• **Initial Condition**

Run the preamble procedure defined in Section 4.2.1 to setup the transport and initiate connection to an Object Server.

• **Test Procedure**

1. The Upper Tester induces the IUT to open an Object Transfer Channel.

2. The IUT sends a _LE_Credit_Based_Connection_Request_ with the PSM parameter set to «PSM_OTS» and other parameters set to valid values.

3. After receiving the _LE_Credit_Based_Connection_Request_, the Lower Tester sends a _LE_Credit_Based_Connection_Rsp_.

• **Expected Outcome**

   **Pass verdict**

   The IUT sends a correctly formatted _LE_Credit_Based_Connection_Request_ with the PSM set to «PSM_OTS».
4.8.12.2 OTP/CL/OTPP/BV-26-I [Open Object Transfer Channel - BR/EDR transport]

- **Test Purpose**
  Verify that the Object Client IUT can perform the Open Object Transfer Channel procedure over BR/EDR transport.

- **Reference**
  [3] 4.5.10

- **Initial Condition**
  Run the preamble procedure defined in Section 4.2.2 to setup the transport and initiate connection to an Object Server.

- **Test Procedure**
  1. The Upper Tester induces the IUT to open an Object Transfer Channel.
  2. The IUT sends an $L2CAP\_ConnectReq$ with the PSM parameter set to «PSM_OTS» and other parameters set to valid values for Enhanced Retransmission Mode (ERTM).
  3. After receiving the $L2CAP\_ConnectReq$, the Lower Tester sends an $L2CAP\_ConnectRsp$.

- **Expected Outcome**
  **Pass verdict**
  The IUT sends a correctly formatted $L2CAP\_ConnectReq$ with the PSM set to «PSM_OTS».

4.8.13 OTP/CL/OTPP/BV-27-I [Read Object – Abort Read Object Contents]

- **Test Purpose**
  Verify that the Object Client IUT can perform an abort after initiating the Read Object Contents sub-procedure.

- **Reference**
  [3] 4.5.4.2

- **Initial Condition**
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.
The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

Run the preamble procedure defined in 4.2.3 to open an Object Transfer Channel.

The Lower Tester contains at least one valid object with valid content that supports the Read property, and this object is selected.

- Test Procedure

1. The Upper Tester induces the IUT to read the part or the whole of the contents of the Current Object.

2. The IUT reads the Object Size characteristic (see OTP/CL/OTR/BV-04-I [Read Object Size Characteristic] in Section 4.5.4).

3. The IUT writes the OACP Read Op Code (0x05) to the OACP with the Offset parameter set to a value less than the Current Size field read in step #2 and the Length Parameter set to a value such that the sum of the Length and Offset parameters is less than or equal to the Current Size field read in step #2.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x05) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

5. The IUT sends an ATT_Handle_Value_Confirmation.

6. The Lower Tester begins to send the requested object contents via the Object Transfer Channel.

7. Before the object transfer has completed, the Lower Tester closes the Object Transfer Channel in order to induce the IUT to abort the object transfer.

8. The IUT writes the OACP Abort Op Code (0x07) to the OACP with no parameter.

9. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x07) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

- Expected Outcome

Pass verdict

In step #3, the IUT writes a correctly formatted Read Op Code (0x05) with the Offset parameter less than the Current Size field read in step #2 and a Length parameter such that the sum of the Length and the Offset parameters is less than or equal to the Current Size field read in step #2.
The Offset and Length values written by the IUT are consistent with the request made by the Upper Tester.

After indication of the OACP, the IUT sends an `ATT_Handle_Value_Confirmation` to the Lower Tester.

At step #8, the IUT writes a correctly formatted Abort Op Code (0x07) to the OACP.

Part of the specified object contents are correctly received via the Object Transfer Channel by the IUT.

After the object transfer has been aborted, the IUT returns to a stable state and can process commands normally.

### 4.9 Error Handling Procedures

This test group contains tests to verify compliant operation when a variety of error conditions occur relating to OACP, OLCP, ATT and other operations.

#### 4.9.1 OACP Error Handling Procedures

This test group contains test cases to verify compliant operation when an OACP error is reported by the Object Server.

#### 4.9.1.1 OTP/CL/OAE/BI-01-I [OACP Error - Op Code Not Supported]

- **Test Purpose**
  
  Verify that the Object Client IUT behaves appropriately when it receives an ‘Op Code Not Supported’ OACP Result Code.

- **Reference**

  [3] 4.6

- **Initial Condition**

  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  The OTS Feature characteristic exposed by the Lower Tester has a value that indicates that the OACP Op Code used in the test procedure is supported. This is done for test cases so that the IUT can be induced to use the Op Code.
Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- Test Procedure

1. The Upper Tester induces the IUT to write any optional Op Code to the OACP.

2. The Lower Tester sends an indication of the OACP characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Op Code Not Supported’ (0x02) (i.e. the Lower Tester simulates that it does not support the Op Code).

3. The IUT sends an **ATT_Handle_Value_Confirmation**.

- Expected Outcome

**Pass verdict**

After receiving the error response, the IUT returns to a stable state and can process commands normally.

### 4.9.1.2 OTP/CL/OAE/BI-02-I [OACP Error – Insufficient Resources]

- Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives an ‘Insufficient Resources’ OACP Result Code.

- Reference

[3] 4.6

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

The Lower Tester contains at least one valid object with read and write properties and this object is selected.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- Test Procedure
1. The Upper Tester induces the IUT to write the OACP Create (0x01), Read (0x05) or Write (0x06) Op Code to the OACP with valid parameter(s).

2. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Insufficient resources’ (0x04) (i.e. the Lower Tester simulates that it does not have sufficient resources to service the request).

3. The IUT sends an ATT_Handle_Value_Confirmation.

- Expected Outcome

**Pass verdict**

If the OACP Op Code used in the test procedure is OACP Read or OACP Write, the IUT opens an Object Transfer Channel before writing the Op Code to the OACP during step #1. After receiving the error response, the IUT returns to a stable state and can process commands normally.

4.9.1.3 OTP/CL/OAE/BI-03-I [OACP Error – Invalid Object]

- Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives an ‘Invalid Object’ OACP Result Code.

- Reference

[3] 4.6

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- Test Procedure

1. The Upper Tester induces the IUT to write the OACP Delete (0x02), Calculate Checksum (0x03), Execute (0x04), Read (0x05) or Write (0x06) Op Code to the OACP.
2. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Invalid Object’ (0x05) (i.e. the Lower Tester simulates that the Current Object is an Invalid Object).

3. The IUT sends an `ATT_Handle_Value_Confirmation`.

   • Expected Outcome

   **Pass verdict**

   If the OACP Op Code used in the test procedure is OACP Read or OACP Write, the IUT opens an Object Transfer Channel before writing the Op Code to the OACP during step #1. After receiving the error response, the IUT returns to a stable state and can process commands normally.

**4.9.1.4 OTP/CL/OAE/BI-04-I [OACP Error – Channel Unavailable]**

   • Test Purpose

   Verify that the Object Client IUT behaves appropriately when it receives a ‘Channel Unavailable’ OACP Result Code.

   • Reference

   [3] 4.6

   • Initial Condition

   Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

   The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

   The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

   The Lower Tester contains at least one valid object with read and write properties and this object is selected.

   Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

   • Test Procedure

   1. The Upper Tester induces the IUT to write the Read (0x05) or Write (0x06) Op Code to the OACP with valid parameter(s).

   2. The Lower Tester closes any Object Transfer Channel that is open.
3. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Channel Unavailable’ (0x06) (i.e. the Lower Tester responds that there is no Object Transfer Channel open).

4. The IUT sends an `ATT_Handle_Value_Confirmation`.

   **Expected Outcome**

   **Pass verdict**

   The IUT opens an Object Transfer Channel before writing the Op Code to the OACP during step #1.

   After receiving the error response, the IUT returns to a stable state and can process commands normally.

4.9.1.5 OTP/CL/OAE/BI-05-I [OACP Error – Unsupported Type]

   **Test Purpose**

   Verify that the Object Client IUT behaves appropriately when it receives an ‘Unsupported Type’ OACP Result Code.

   **Reference**

   [3] 4.6

   **Initial Condition**

   Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

   The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

   The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

   Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

   **Test Procedure**

   1. The Upper Tester induces the IUT to write the OACP Create (0x01) Op Code to the OACP with valid parameter(s).

   2. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code
followed by the Result Code value for ‘Unsupported Type’ (0x07) (i.e. the Lower Tester simulates that it does not support this object type).

3. The IUT sends an $\textit{ATT\_Handle\_Value\_Confirmation}$.

- Expected Outcome

**Pass verdict**

After receiving the error response, the IUT returns to a stable state and can process commands normally.

### 4.9.1.6 OTP/CL/OAE/BI-06-I [OACP Error – Procedure Not Permitted]

- **Test Purpose**

Verify that the Object Client IUT behaves appropriately when it receives a ‘Procedure Not Permitted’ OACP Result Code.

- **Reference**

[3] 4.6

- **Initial Condition**

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

The Lower Tester contains at least one valid object with read and write, delete and execute properties supported and this object is selected.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- **Test Procedure**

1. The Upper Tester induces the IUT write the OACP Delete (0x02), Execute (0x04), Read (0x05) or Write (0x06) Op Code to the OACP with valid parameter(s).

2. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Procedure Not Permitted’ (0x08) (i.e. the Lower Tester simulates that the object’s properties does not permit the given operation).

3. The IUT sends an $\textit{ATT\_Handle\_Value\_Confirmation}$.
• Expected Outcome

**Pass verdict**

If the OACP Op Code used in the test procedure is OACP Read or OACP Write, the IUT opens an Object Transfer Channel before writing the Op Code to the OACP during step #1. After receiving the error response, the IUT returns to a stable state and can process commands normally.

4.9.1.7 OTP/CL/OAE/BI-07-I [OACP Error – Object Locked]

• Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives an ‘Object Locked’ OACP Result Code.

• Reference

[3] 4.6

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

The Lower Tester contains at least one valid object with read and write properties supported and this object is selected.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

• Test Procedure

1. The Upper Tester induces the IUT write the OACP Read (0x05), Write (0x06), Execute (0x04), Delete (0x02) or Calculate Checksum (0x03) Op Code to the OACP with valid parameter(s).

2. The Lower Tester sends an indication of the OACP Control Point characteristic with the Response Code Op Code (0x60) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Object Locked’ (0x09) (i.e. the Lower Tester simulates that the object is locked).

3. The IUT sends an **ATT_Handle_Value_Confirmation**.
• Expected Outcome

Pass verdict
If the OACP Op Code used in the test procedure is OACP Read or OACP Write, the IUT opens an Object Transfer Channel before writing the Op Code to the OACP during step #1. After receiving the error response, the IUT returns to a stable state and can process commands normally.

4.9.1.8 OTP/CL/OAE/BI-08-I [OACP Error – Operation Failed]

• Test Purpose
Verify that the Object Client IUT behaves appropriately when it receives an ‘Operation Failed’ OACP Result Code.

• Reference
[3] 4.6

• Initial Condition
Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.
The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.
The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.
Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

• Test Procedure
1. The Upper Tester induces the IUT write any Op Code to the OACP using an appropriate Parameter for the Op Code.
2. The Lower Tester sends an indication of the OACP characteristic with the Response Code Op Code (0x60) and a Parameter representing Request Op Code followed by the Response Code Value for ‘Operation Failed’ (0x0A) (i.e. the Lower Tester simulates a failed operation).

• Expected Outcome
Pass verdict
After receiving the error response, the IUT returns to a stable state and can process commands normally.
4.9.1.9 OTP/CL/OAE/BI-09-I [OACP Error – Procedure Timeout]

- Test Purpose

Verify that when the Object Client IUT does not receive an indication of the OACP characteristic in response to an Op Code, it times out after 30 seconds.

- Reference

[3] 4.4.4

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

- Test Procedure

1. The Upper Tester induces the IUT to write any of the supported Op Codes to the control point.

2. The IUT writes an Op Code to the OACP, along with an appropriate Parameter if one is required by the Op Code.

3. The Lower Tester does not send an indication of the OACP characteristic for at least 30 seconds.

4. After the specified timeout has expired, the IUT notifies the Upper Tester that the procedure timeout has expired and the IUT considers the procedure to have failed.

5. The Upper Tester attempts to induce the IUT to perform an additional OACP operation.

- Expected Outcome

Pass verdict

In step #5, the IUT either does not write to the OACP, or establishes a new ATT Bearer before the operation is performed.

The IUT returns to a stable state in all cases.
### 4.9.2 OLCP Error Handling Procedures

This test group contains test cases to verify compliant operation when an OLCP error is reported by the Object Server.

#### 4.9.2.1 OTP/CL/OLE/BI-01-I [OLCP Error – Op Code Not Supported]

- **Test Purpose**
  
  Verify that the Object Client IUT behaves appropriately when it receives an ‘Op Code Not Supported’ OLCP Result Code.

- **Reference**
  
  [3] 4.4.2.1

- **Initial Condition**
  
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

  The OTS Feature characteristic exposed by the Lower Tester has a value that indicates that the OLCP Op Code used in the test procedure is supported. This is done for test cases so that the IUT can be induced to use the Op Code.

- **Test Procedure**

  1. The Upper Tester induces the IUT to write any optional Op Code to the OLCP.

  2. The Lower Tester sends an indication of the OLCP characteristic with the Response Code Op Code (0x70) and a Parameter representing the Request Op Code followed by the Result Code value for ‘Op Code Not Supported’ (0x02) (i.e. the Lower Tester simulates that it does not support the Op Code).

  3. The IUT sends an `ATT_Handle_Value_Confirmation`.

- **Expected Outcome**

  **Pass verdict**

  After receiving the error response, the IUT returns to a stable state and can process commands normally.
4.9.2.2 OTP/CL/OLE/BI-02-I [OLCP Error – Operation Failed]

- Test Purpose
  Verify that the Object Client IUT behaves appropriately when it receives an 'Operation Failed' OLCP Result Code.

- Reference
  [3] 4.4.2.1

- Initial Condition
  Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

  The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

  The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

  Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

- Test Procedure
  1. The Upper Tester induces the IUT to write a valid OLCP Op Code to the OLCP with valid parameter(s).

  2. The Lower Tester sends an indication of the OLCP characteristic with the Response Code Op Code (0x70) and a Parameter representing Request Op Code followed by the Response Code Value for ‘Operation Failed’ (0x04) (i.e. the Lower Tester simulates a failed operation).

- Expected Outcome
  Pass verdict
  The IUT returns to a stable state and can process commands normally.

4.9.2.3 OTP/CL/OLE/BI-03-I [OLCP Error – Out of Bounds]

- Test Purpose
  Verify that the Object Client IUT behaves appropriately when it receives an 'Out of Bounds' OLCP Result Code.
Object Transfer Profile (OTP)
Bluetooth Test Specification

• Reference

[3] 4.4.2.1

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

• Test Procedure

1. The Upper Tester induces the IUT to write the OLCP Next (0x04) to the OLCP without parameters.

2. The Lower Tester sends an indication of the OLCP characteristic with the Response Code Op Code (0x70) and a Parameter representing Request Op Code followed by the Response Code Value for ‘Out Of Bounds’ (0x05) (i.e. the Lower Tester simulates that the requested procedure attempted to select an object beyond the last object in the current list).

• Expected Outcome

Pass verdict
The IUT returns to a stable state and can process commands normally.

4.9.2.4 OTP/CL/OLE/BI-04-I [OLCP Error – Too Many Objects]

• Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives a ‘Too Many Objects’ OLCP Result Code.
• Reference

[3] 4.4.2.1

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

• Test Procedure

1. The Upper Tester induces the IUT to write a valid OLCP Op Code to the OLCP with valid parameter(s).

2. The Lower Tester sends an indication of the OLCP characteristic with the Response Code Op Code (0x70) and a Parameter representing Request Op Code followed by the Response Code Value for ‘Too Many Objects’ (0x06) (i.e. the Lower Tester simulates that there are too many objects in the current list).

• Expected Outcome

Pass verdict

The IUT returns to a stable state and can process commands normally.

4.9.2.5  OTP/CL/OLE/BI-05-I [OLCP Error – No Object]

• Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives a ‘No Object’ OLCP Result Code.

• Reference

[3] 4.4.2.1

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.
The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.

- Test Procedure

1. The Upper Tester induces the IUT to write a valid OLCP Op Code to the OLCP with valid parameter(s).

2. The Lower Tester sends an indication of the OLCP characteristic with the Response Code Op Code (0x70) and a Parameter representing Request Op Code followed by the Response Code Value for ‘No Object’ (0x07) (i.e. the Lower Tester simulates that there are no objects in the current list).

- Expected Outcome

Pass verdict

The IUT returns to a stable state and can process commands normally.

4.9.2.6 OTP/CL/OLE/BI-06-1 [OLCP Error – Object ID Not Found]

- Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives an ‘Object ID Not Found’ OLCP Result Code.

- Reference

[3] 4.4.2.1

- Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OLCP for indications.
The IUT has previously discovered the object IDs of the Lower Tester by performing an
Object Discovery procedure (see OTP/CL/OTPP/BV-01-I [Object Discovery – Discover All
Objects] in Section 4.8.1 or OTP/CL/OTPP/BV-13-I [Object Discovery - Discover by
Directory Listing Object] in Section 4.8.4).

• Test Procedure

1. The Upper Tester induces the IUT to write the OLCP Go To (0x05) Op Code to the
   OLCP with any object ID (other than a RFU value or 0x000000000000) as parameter.
2. The Lower Tester sends an indication of the OLCP characteristic with the Response
   Code Op Code (0x70) and a Parameter representing Request Op Code followed by the
   Response Code Value for ‘Object ID Not Found’ (0x08) (i.e. the Lower Tester simulates
   that the object ID cannot be found).

• Expected Outcome

Pass verdict
The IUT returns to a stable state and can process commands normally.

4.9.2.7 OTP/CL/OLE/BI-07-I [OLCP Error – Procedure Timeout]

• Test Purpose

Verify that when the Object Client IUT does not receive an indication of the OLCP
characteristic in response to an Op Code, it times out after 30 seconds.

• Reference

[3] 4.4.4

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if
using a BR/EDR transport, to setup the transport and initiate connection to an Object
Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all
defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the
Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic
Configuration for Indication] to configure the OLCP for indications.

• Test Procedure

1. The Upper Tester induces the IUT to write any of the supported Op Codes to the control
   point.
2. The IUT writes an Op Code to the OLCP, along with an appropriate Parameter if one is required by the Op Code.

3. The Lower Tester does not send an indication of the OLCP characteristic for at least 30 seconds.

4. After the specified timeout has expired, the IUT notifies the Upper Tester that the procedure timeout has expired and the IUT considers the procedure to have failed.

5. The Upper Tester attempts to induce the IUT to perform an additional OLCP operation.

   • Expected Outcome

   Pass verdict
   In step #5, the IUT either does not write to the OLCP, or establishes a new ATT Bearer before the operation is performed.
   The IUT returns to a stable state in all cases.

4.9.3 Application Error Handling Procedures

This test group contains test cases to verify compliant operation when an Attribute Protocol Application Error Code is reported by the Object Server.

4.9.3.1 OTP/CL/APE/BI-01-I [Application Error – Concurrency Limit Exceeded]

   • Test Purpose

   Verify that the Object Client IUT behaves appropriately when it receives a 'Concurrency Limit Exceeded' Attribute Protocol Application Error Code.

   • Reference

[3] 4.6

   • Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

The Lower Tester contains at least 3 valid objects.
• Test Procedure

1. The Upper Tester induces the IUT to write the OLCP First Op Code (0x01) to the OLCP without a Parameter Value to select the first object.

2. The Lower Tester responds with a ‘Concurrency Limit Exceeded’ Attribute Protocol Application Error Code.

3. The Upper Tester again induces the IUT to write the OLCP First Op Code (0x01) to the OLCP without a Parameter Value to select the first object.

4. The Lower Tester, after sending a Write Response to acknowledge the write to the OLCP, sends an indication of the OLCP characteristic containing the Response Code Op Code (0x70), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

• Expected Outcome

Pass verdict

In step #2, the IUT returns to a stable state and can process commands normally.

In step #4, after receiving each indication of the OLCP, the IUT sends an ATT_Handle_Value_Confirmation to the Lower Tester.

4.9.3.2 OTP/CL/APE/BI-02-I [Application Error – Object Name Already Exists]

• Test Purpose

Verify that the Object Client IUT behaves appropriately when it receives an ‘Object Name Already Exists’ Attribute Protocol Application Error Code.

• Reference

[3] 4.6

• Initial Condition

Run the preamble procedure defined in Section 4.2.1, if using an LE transport, or 4.2.2, if using a BR/EDR transport, to setup the transport and initiate connection to an Object Server.

The Lower Tester includes one instantiation of the Object Transfer Service [6] including all defined characteristics.

The IUT knows the handles of the Object Transfer Service characteristics contained in the Lower Tester from running the test procedures in Section 4.4 or by other means.

Run the preamble procedure defined in GATT.TS [5] Section 4.2.2.2 [Characteristic Configuration for Indication] to configure the OACP for indications.

The Lower Tester contains at least 3 valid objects.
The Lower Tester does not support the Object Changed characteristic.

• Test Procedure

1. The Upper Tester induces the IUT to create a new object.

2. The IUT writes the OACP Create Op Code (0x01) to the OACP with the Type parameter set to valid object type (e.g., the UUID “Firmware”) and the Size Parameter set to a valid value (e.g., 0x00000400, representing an allocated object size of 1024 octets).

3. The Lower Tester, after sending a Write Response to acknowledge the write to the OACP, sends an indication of the OACP characteristic containing the Response Code Op Code (0x60), the Request Op Code (0x01) followed by the Result Code for ‘Success’ (0x01) without a Response Parameter.

4. The IUT sends an ATT_Handle_Value_Confirmation.

5. The IUT writes the Object Name characteristic with a non-zero length UTF-8 string (see OTP/CL/OTW/BV-01-I [Write Object Name Characteristic] in Section 4.6.1).

6. The Lower Tester creates another object with the same Object Name as the one requested. (This step is included to simulate that the requested name had already been taken by an existing object.)

7. The Lower Tester responds with an ‘Object Name Already Exists’ Attribute Protocol Application Error Code.

8. The Upper Tester provides a new name if requested by the IUT.

9. The IUT writes a different Object Name characteristic with a non-zero length UTF-8 string (see OTP/CL/OTW/BV-01-I [Write Object Name Characteristic] in Section 4.6.1). (It is permitted for the IUT to perform other operations between step #7 and this step to e.g., re-discover the database of objects and their metadata.)

10. The Lower Tester sends a Write Response to acknowledge the write to the Object Name.

• Expected Outcome

Pass verdict

In step #7, the IUT returns to a stable state and can process commands normally.

In step #9, the IUT writes a different Object Name than attempted in step #5.
## 5 Test Case Mapping

The Test Case Mapping Table (TCMT) maps test cases to specific capabilities in the ICS. Profiles, protocols and services may define multiple roles, and it is possible that a product may implement more than one role. The product shall be tested in all roles for which support is declared in the ICS document. For products which support more than one role, a separate TCMT shall be filled out for each role, and separate tests shall be conducted for each role.

The columns for the TCMT are defined as follows:

- **Item**: contains an y/x reference, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS Proforma for OTP [4]. If the item is defined with Protocol, Profile or Service abbreviation before y/x, the table and feature number referenced are defined in the abbreviated ICS proforma document.

- **Feature**: recommended to be the primary feature defined in the ICS being tested or may be the test case name.

- **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.

- **Test Case Applicable**: may be used to note if a test is required based on the supported features.

For purpose and structure of the ICS/IXIT proforma and instructions for completing the ICS/IXIT proforma refer to the Bluetooth ICS and IXIT proforma documents.

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
<th>Test Case Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTP 2/2 AND OTP 3/2 AND (OTP 9/4 OR OTP 9/5)</td>
<td>Discover Object Transfer Service over LE - Primary</td>
<td>OTP/CL/OTSD/BV-01-I</td>
<td></td>
</tr>
<tr>
<td>OTP 2/2 AND OTP 3/2 AND OTP 9/6</td>
<td>Discover Object Transfer Service over LE - Secondary</td>
<td>OTP/CL/OTSD/BV-02-I</td>
<td></td>
</tr>
<tr>
<td>OTP 9/7</td>
<td>SDP Service Discovery</td>
<td>OTP/CL/OTSD/BV-03-I</td>
<td></td>
</tr>
<tr>
<td>(OTP 3/2 AND NOT OTP 3/1) AND OTP 2/1 AND GAP 0/3</td>
<td>Object Transfer Service not discoverable over BR/EDR</td>
<td>OTP/SR/OTSD/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>OTP 4/2</td>
<td>Object Transfer Service UUID in AD</td>
<td>OTP/SR/OTSD/BV-05-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/1</td>
<td>Discover OTS Feature Characteristic</td>
<td>OTP/CL/OTCD/BV-01-I</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td>Test Case Applicable</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OTP 6/2</td>
<td>Discover Object Name Characteristic</td>
<td>OTP/CL/OTCD/BV-02-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/3</td>
<td>Discover Object Type Characteristic</td>
<td>OTP/CL/OTCD/BV-03-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/4</td>
<td>Discover Object Size Characteristic</td>
<td>OTP/CL/OTCD/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/5</td>
<td>Discover Object First-Created Characteristic</td>
<td>OTP/CL/OTCD/BV-05-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/6</td>
<td>Discover Object Last-Modified Characteristic</td>
<td>OTP/CL/OTCD/BV-06-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/7</td>
<td>Discover Object ID Characteristic</td>
<td>OTP/CL/OTCD/BV-07-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/8</td>
<td>Discover Object Properties Characteristic</td>
<td>OTP/CL/OTCD/BV-08-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/9</td>
<td>Discover Object Action Control Point (OACP)</td>
<td>OTP/CL/OTCD/BV-09-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/10</td>
<td>Discover Object List Control Point (OLCP)</td>
<td>OTP/CL/OTCD/BV-10-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/11</td>
<td>Discover Object List Filter Characteristics</td>
<td>OTP/CL/OTCD/BV-11-I</td>
<td></td>
</tr>
<tr>
<td>OTP 6/12</td>
<td>Discover Object Changed Characteristic</td>
<td>OTP/CL/OTCD/BV-12-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/1</td>
<td>Read OTS Feature Characteristic</td>
<td>OTP/CL/OTCD/BV-01-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/2</td>
<td>Read Object Name Characteristic</td>
<td>OTP/CL/OTCD/BV-02-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/3</td>
<td>Read Object Type Characteristic</td>
<td>OTP/CL/OTCD/BV-03-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/4</td>
<td>Read Object Size Characteristic</td>
<td>OTP/CL/OTCD/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/5</td>
<td>Read Object First-Created Characteristic</td>
<td>OTP/CL/OTCD/BV-05-I</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td>Test Case Applicable</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------</td>
<td>---------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OTP 7/10</td>
<td>Read Object Last-Modified Characteristic</td>
<td>OTP/CL/OTR/BV-06-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/12</td>
<td>Read Object ID Characteristic</td>
<td>OTP/CL/OTR/BV-07-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/13</td>
<td>Read Object Properties Characteristic</td>
<td>OTP/CL/OTR/BV-08-I</td>
<td>OTP/CL/OTR/BI-02-I</td>
</tr>
<tr>
<td>OTP 7/19</td>
<td>Read Object List Filter Characteristic</td>
<td>OTP/CL/OTR/BV-09-I</td>
<td>OTP/CL/OTR/BI-03-I</td>
</tr>
<tr>
<td>OTP 7/3</td>
<td>Read Long Object Name Characteristic</td>
<td>OTP/CL/OTR/BV-10-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/22</td>
<td>Read Long Object List Filter Characteristic</td>
<td>OTP/CL/OTR/BV-11-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/4</td>
<td>Write Object Name Characteristic</td>
<td>OTP/CL/OTW/BV-01-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/9</td>
<td>Write Object First-Created Characteristic</td>
<td>OTP/CL/OTW/BV-02-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/11</td>
<td>Write Object Last-Modified Characteristic</td>
<td>OTP/CL/OTW/BV-03-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/14</td>
<td>Write Object Properties Characteristic</td>
<td>OTP/CL/OTW/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/20</td>
<td>Write Object List Filter Characteristic</td>
<td>OTP/CL/OTW/BV-05-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/5</td>
<td>Write Long Object Name Characteristic</td>
<td>OTP/CL/OTW/BV-06-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/21</td>
<td>Write Long Object List Filter Characteristic</td>
<td>OTP/CL/OTW/BV-07-I</td>
<td></td>
</tr>
<tr>
<td>OTP 7/23</td>
<td>Object Changed</td>
<td>OTP/CL/OTI/BV-01-I</td>
<td>OTP/CL/OTI/BV-02-I</td>
</tr>
<tr>
<td>OTP 7/21</td>
<td>Service Changed</td>
<td>OTP/CL/OTI/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td>Test Case Applicable</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OTP 8/2</td>
<td>Object Discovery - Discover All Objects</td>
<td>OTP/CL/OTPP/BV-01-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/3 OR OTP 8/16</td>
<td>Object Discovery - Search for Specific Object</td>
<td>OTP/CL/OTPP/BV-02-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/5</td>
<td>Object Discovery - Discover by Filter - Name Starts With</td>
<td>OTP/CL/OTPP/BV-03-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/6</td>
<td>Object Discovery - Discover by Filter - Name Ends With</td>
<td>OTP/CL/OTPP/BV-04-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/7</td>
<td>Object Discovery - Discover by Filter - Name Contains</td>
<td>OTP/CL/OTPP/BV-05-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/8</td>
<td>Object Discovery - Discover by Filter - Name is Exactly</td>
<td>OTP/CL/OTPP/BV-06-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/9</td>
<td>Object Discovery - Discover by Filter - Object Type</td>
<td>OTP/CL/OTPP/BV-07-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/10</td>
<td>Object Discovery - Discover by Filter - Created between</td>
<td>OTP/CL/OTPP/BV-08-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/11</td>
<td>Object Discovery - Discover by Filter - Modified between</td>
<td>OTP/CL/OTPP/BV-09-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/12</td>
<td>Object Discovery - Discover by Filter - Current Size between</td>
<td>OTP/CL/OTPP/BV-10-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/13</td>
<td>Object Discovery - Discover by Filter - Allocated Size between</td>
<td>OTP/CL/OTPP/BV-11-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/14</td>
<td>Object Discovery - Discover by Filter - Marked Objects</td>
<td>OTP/CL/OTPP/BV-12-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/15</td>
<td>Object Discovery - Discover by Directory Listing Object</td>
<td>OTP/CL/OTPP/BV-13-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/17</td>
<td>Select Object - Select by Object ID</td>
<td>OTP/CL/OTPP/BV-14-I OTP/CL/OLE/BI-06-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/19</td>
<td>Read Object - Read Object Contents</td>
<td>OTP/CL/OTPP/BV-15-I</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td>Test Case Applicable</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OTP 8/19 AND OTP 8/29</td>
<td>Abort Read Object Contents</td>
<td>OTP/CL/OTPP/BV-27-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/20 AND OTP 7/5</td>
<td>Write Object - Write Object Metadata</td>
<td>OTP/CL/OTPP/BV-16-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/21</td>
<td>Write Object - Write Object Contents</td>
<td>OTP/CL/OTPP/BV-17-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/21 AND OTS 8/31</td>
<td>Write Object - Write Object Contents - Truncate</td>
<td>OTP/CL/OTPP/BV-18-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/22 AND OTP 8/27</td>
<td>Resume Reading Object Contents</td>
<td>OTP/CL/OTPP/BV-19-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/23</td>
<td>Resume Writing Object Contents - Current Size Method</td>
<td>OTP/CL/OTPP/BV-20-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/24 AND OTP 8/27</td>
<td>Resume Writing Object Contents - Data Integrity Method</td>
<td>OTP/CL/OTPP/BV-21-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/25</td>
<td>Create Object</td>
<td>OTP/CL/OTPP/BV-22-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/26</td>
<td>Delete Object</td>
<td>OTP/CL/OTPP/BV-23-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/28</td>
<td>Execute Object</td>
<td>OTP/CL/OTPP/BV-24-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/30 AND OTP 3/2</td>
<td>Open Object Transfer Channel LE transport</td>
<td>OTP/CL/OTPP/BV-25-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/30 AND OTP 3/1</td>
<td>Open Object Transfer Channel BR/EDR transport</td>
<td>OTP/CL/OTPP/BV-26-I</td>
<td></td>
</tr>
<tr>
<td>OTP 8/25</td>
<td>OACP Create Errors</td>
<td>OTP/CL/OAE/BI-05-I OTP/CL/APE/BI-02-I</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
<td>Test Case Applicable</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OTP 7/17 AND OTP 8/7</td>
<td>OLCP Op Code Not Supported</td>
<td>OTP/CL/OLE/BI-01-I</td>
<td></td>
</tr>
</tbody>
</table>

*Table 5.1: Test Case Mapping*