|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* |  |
| Source:\* |  |
| Date:\* |  |
| Reason for Change/s:\* |  |
| CR against: Release\* |  |
| CR against: WI\* | Active <#WI#>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number:  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* |  |
| Clauses \* |  |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Other TS/TR(s) impacted | None |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2019 (do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

#REASON#

<https://git.onem2m.org/specifications/ts/ts-0019/-/merge_requests/38>

<https://git.onem2m.org/specifications/ts/ts-0019/-/merge_requests/38/diffs?commit_id=8db42191a9118452017dff4de0c644371374a660>

----------------------- Start of change 1 -----------------------



| **oneM2M** **Technical** **Specification** | **oneM2M** **Technical** **Specification** |
| --- | --- |
|  |  |
| Document Number | oneM2M-TS-0019-V-4\_1.1 |
| Document Name: | Abstract Test Suite and Implementation eXtra Information for Test |
| Date: | 2024 March 1 |
| Abstract: | Abstract Test Suite and Implementation eXtra Information for Test consists of : - Definition of the Abstract Protocol Tester (APT) - Definition of TTCN-3 test architecture - Development of TTCN-3 test suite, e.g. naming conventions, code documentation, test case structure. - IXIT proforma;\ |

|  |  |
| --- | --- |
| Date: | 2024 December 1 |
| Abstract: | Abstract Test Suite and Implementation eXtra Information for Test consists of:   * Definition of the Abstract Protocol Tester (APT) * Definition of TTCN-3 test architecture * Development of TTCN-3 test suite, e.g. naming conventions, code documentation, test case structure. * IXIT proforma; |
| Template Version:23 February 2015 (Do not modify) | |

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners’ Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Copyright Notification

1. 2025, oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC).

All rights reserved.

The copyright extends to reproduction in all media.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

----------------------- End of change 1 -----------------------

----------------------- Start of change 2 -----------------------

## 3.2 Symbols

Void.

My new change

----------------------- End of change 2 -----------------------

----------------------- Start of change 3 -----------------------

## 5.3 Test architecture

The approach for the implementation of an Abstract Protocol Tester selected in oneM2M follows the recommendation of the oneM2M Testing Framework oneM2M TS-0015 [i.2] where the TTCN-3 language and its architecture are recommended.

Following this recommendation the oneM2M tester architecture comprises a non-platform dependent Test Suite, and a platform dependent part.

 > NOTE: However, it can be implemented in a semi-independent manner, which will minimize the dependency to those elements.

* **oneM2M** **TTCN** **-3** **Abstract** **Test Suite:** the test suite is platform independent, and it is the cornerstone of the architecture. It allows a complete decoupling between the test suite and the rest of the test system. The test suite is composed of a complete set of test cases covering oneM2M requirements specified by oneM2M TS0001 [1] and oneM2M TS-0004 [2].
* **oneM2M System Adaptor** **:** this is the platform dependent part that includes adaptors and codecs (out of the scope of the present document). This part of the architecture definition depends on the specific platform (e.g. Windows® or Linux®) and test tool on which the tester is going to run.

Figure 5.3-2 shows the oneM2M TTCN-3 test architecture design used for the oneM2M ATS. The Test Suite needs to interact with the System Adaptor to implement the collection of TTCN-3 test cases that are intended to be used to test the oneM2M IUTs.

The oneM2M TTCN-3 test cases implement the test algorithms specified in the TSS&TP document oneM2M TS0018 [7], including verdict logic that allows pass/fail diagnosis.

The test algorithms use the interfaces defined in [1] and [2] (mca, mcc) in order to:

1. control the test event to be sent towards the IUT; and
2. observe the test events received from the IUT.

In TTCN-3 these two interfaces have been implemented through a set of logical TTCN-3 ports (mcaPort and mcaPortIn for mca interface, and mccPort and mccPortIn for mcc interface) which allows oneM2M message primitives exchange with the IUT.

As figure 5.1-1 illustrates, the corresponding ATS needs to use lower layers to establish a proper connection to the implementation under test (SUT) over several physical links (Lower layers link). Four different lower layers have been specified corresponding to the binding protocols considered in oneM2M: HTTP, CoAP, WebSocket and MQTT.

This is a new sentence.

----------------------- End of change 3 -----------------------