# Requirements-driven test case development: a case study

Within the scope of STF 433 a case study of requirements-driven test case development was performed.

Requirements-driven test case development refers to the approach when test development is based on requirements elicited from target system documentation. This implies:

1. requirement elicitation: requirements need to be identified in the ETSI TTCN-3 standard and stored as named entities;
2. test purpose development: requirements are refined into fine-grained test purposes that informally specify situations to be tested; each test purpose targets exactly one requirement;
3. test case development: test case has explicit attribution to the test purpose or requirement it implements; such attribution allows for traceability matrix and requirement-based coverage reporting.

Within the STF 433 the target system (system under test, SUT) is a TTCN-3 processor, either compiler or interpreter. The documentation is represented by TTCN-3 Standard Part 1 " Core Language"[[1]](#footnote-1).

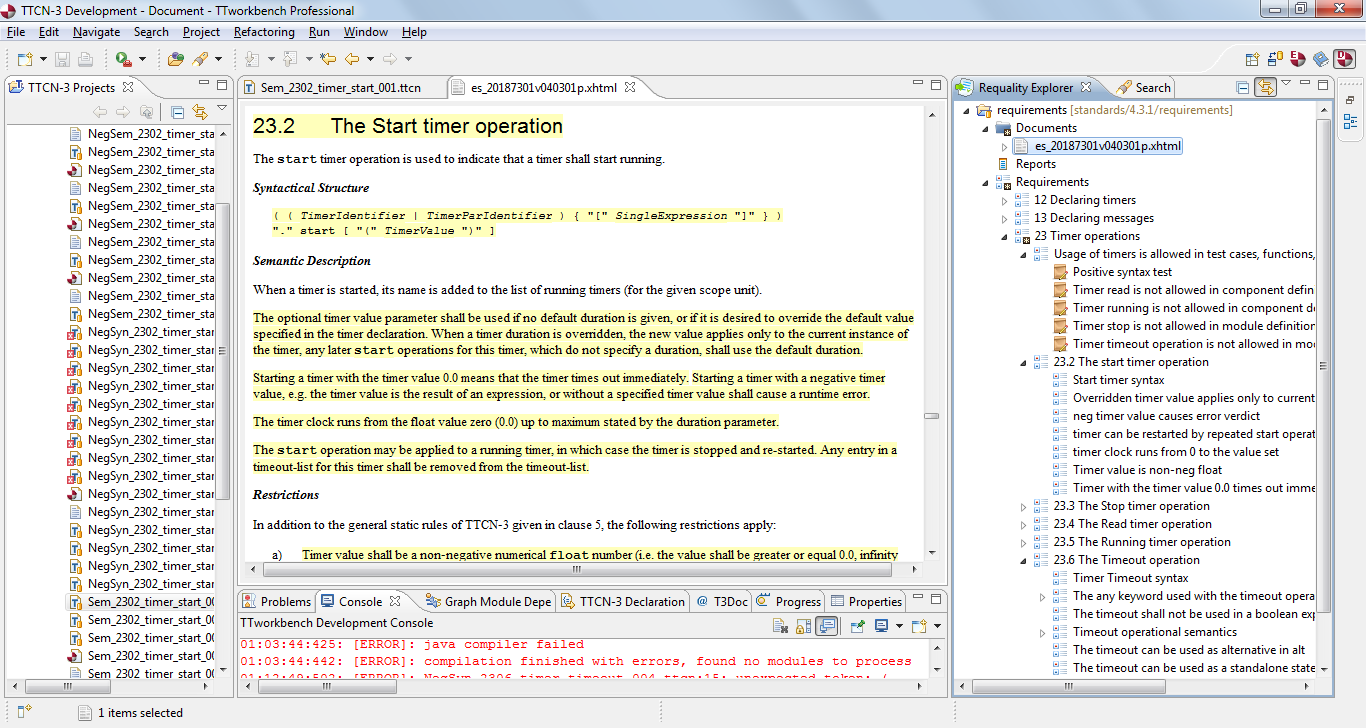
The development process established in the previous STF 409 included requirements traceability only partially: each test case has a reference to a section of the standard. Since sections of the standard are typically one page long such reference makes detailed coverage analysis hardly possible since it requires experts read through big chunks of text to evaluate the test case and identify covered requirements. The task of identification of missing requirements is even more challenging since there are no visual or other easily recognizable marks that denotate requirements.

To elicit requirements and develop test purposes we used Requality tool being developed by ISPRAS[[2]](#footnote-2). It is an Eclipse plugin that provides a framework for requirements-based test development:

1. Document markup to highlight requirements. This feature simplifies evaluation completeness of the requirements set.
2. Hierarchical requirements management suitable for SVN/GIT version control systems.
3. Test purpose management. Test purposes are attached as leaves to requirements tree. Test purposes specify desired SUT state, test action and expected output.
4. Reporting facilities.
5. Change management (under development). The feature is designed to track changes between subsequent versions of the documentation and identify modified, removed and new requirements.

Requality could be installed into Eclipse 3.6 and higher from the update site

http://forge.ispras.ru/repo/requality/site\_daily/



Requirement

Test Purpose

Mapping to test case

Figure 1 Requality in TTWorkbench setup

## The case study

Within the case study STF 433 evaluated feasibility of requirements-driven approach to TTCN-3 ATS development. The scope of the case study comprised of 4 chapters:

* Chapter 12 "Declaring messages",
* Chapter 13 "Declaring timers",
* Chapter 23 "Timer operations",
* Chapter 24 "Verdict operations";

The selected chapters are 7,5 pages long in total. 56 requirements were elicited. 7 requirements refined into 34 test purposes; the requirements and test purposes were covered by 174 test cases. Tracking between test cases and requirements/test purposes was established by means of special comments in the TTCN-3 test cases:

// #reqname /Requirements/23 Timer operations/23.2 The start timer operation/Timer with the timer value 0.0 times out immediately.

The string starting with /Requirements is the full name of the requirement or test purpose in the requirements tree. Nikolay Pakulin (ISPRAS) implemented report generator to track requirements coverage based on those comment string.

The detailed report on requirements and test purpose coverage is in the annex.

The case study showed that requirements-based test development is feasible for TTCN-3 ATS development.

## Test purposes

The test purposes were developed using ExTRA[[3]](#footnote-3) (TPLan v.2) notation. The notation was designed for defining test purposes in telecom testing, and its adoption for TTCN3 tool testing required some effort.

First of all, TPLan is implied to describe relatively simple data structures, such as protocol messages. As such, it includes statements like

when {

IUT receives a message

containing field ABC

indicating come value

}

**Listing 1.** Sample ExTRA specification of *when* clause for a fictional protocol

In TTCN3 testing the IUT is a TTCN3 processor (interpreter or compiler) and the input is not a packet but a TTCN3 script with much more complicated structure.

    when {  
        IUT receives TTCN3 module  
        defining component PTC {  
            defining port p  
        }  
        defining component MTC {  
            defining port p  
        }  
          
        defining test case TC {  
            -- contents of template T is not specified here  
            defining template T  
            and creating component PTC  
            and connecting MTC.p with PTC.p  
            and starting component PTC  
                with function {  
                    invoking alt statement {  
                        branch p.receive with setverdict(pass)  
                        branch timeout with setverdict(fail)  
                    }  
                }  
            -- Sending the local template  
            and invoking MTC.p.send(T)  
        }  
    }

**Listing 2.** ExTRA specification of *when* clause for one of a STF 433 test purposes

As Listing 2 shows, test purpose specification in STF 433 is close to the structure of the resulting TTCN3 test case. It prescribes to define MTC and PTC, describes the contents of the test case and the sequence of operations. The test purpose language ExTRA does not provide statements for complex syntax structures like TTCN3 definitions and instructions. In order to present constraints on the TTCN3 text we introduced new verb *define* and new nouns *component*, *altstep*, *function*, *test case*.

To present the internal structure of the TTCN3 specification elements, such as test case, component, etc., we extended the ExTRA language with blocks in curly braces *{* and *}*. The blocks contain sequences of predicates to sub-elements. The predicates are in the form of present participle (verb + ing) and object (noun) with optional identifier and adjectives qualifying the noun. Example: *creating component PTC*. The verbs are *define*, *connect*, *invoke*. The predicates might have optional *with* clause that specifies extra constraints on an element.

    then {  
        IUT accepts the module  
        and IUT executes TC  
            with verdict pass  
    }

**Listing 3.** ExTRA specification of assertionfor one of a STF 433 test purposes

To specify assertion part of *ensure that* clause we introduced verbs *accept*, *reject* and *execute*. The verb *accept* asserts that the input TTCN3 module is syntactically valid and compilation (if any) finishes successfully. The verb *reject* asserts that the input should not pass compilation or fail execution with *error* verdict. The verb *execute* asserts the result of a test case execution in *with verdict* clause.

The test purposes were specified in Requality; still the tool does not have a specific support for ExTRA – no syntax highlighting, validation or test skeleton generation. The tool considered the test purposes as plain unstructured text to store in the database.

# Coverage report generation

The coverage report presented in Annex 3 was generated by ReqsTracer report generator being developed by ISPRAS.

The report generator is an Eclipse plugin. It could be installed from the update site <http://forge.ispras.ru/repo/reqstracer/site/>

At the moment the report generator is a prototype rather than fully functional tool.

To use the report generator, switch to **Requality perspective**. In **Requality Explorer**navigate to the root of a requirement tree that you want to include in the coverage. It might be the root *Requirements* node or any of its child node (except for Test Purpose nodes). Right click on the selected node and choose menu option *Generate ReqTracer report*.

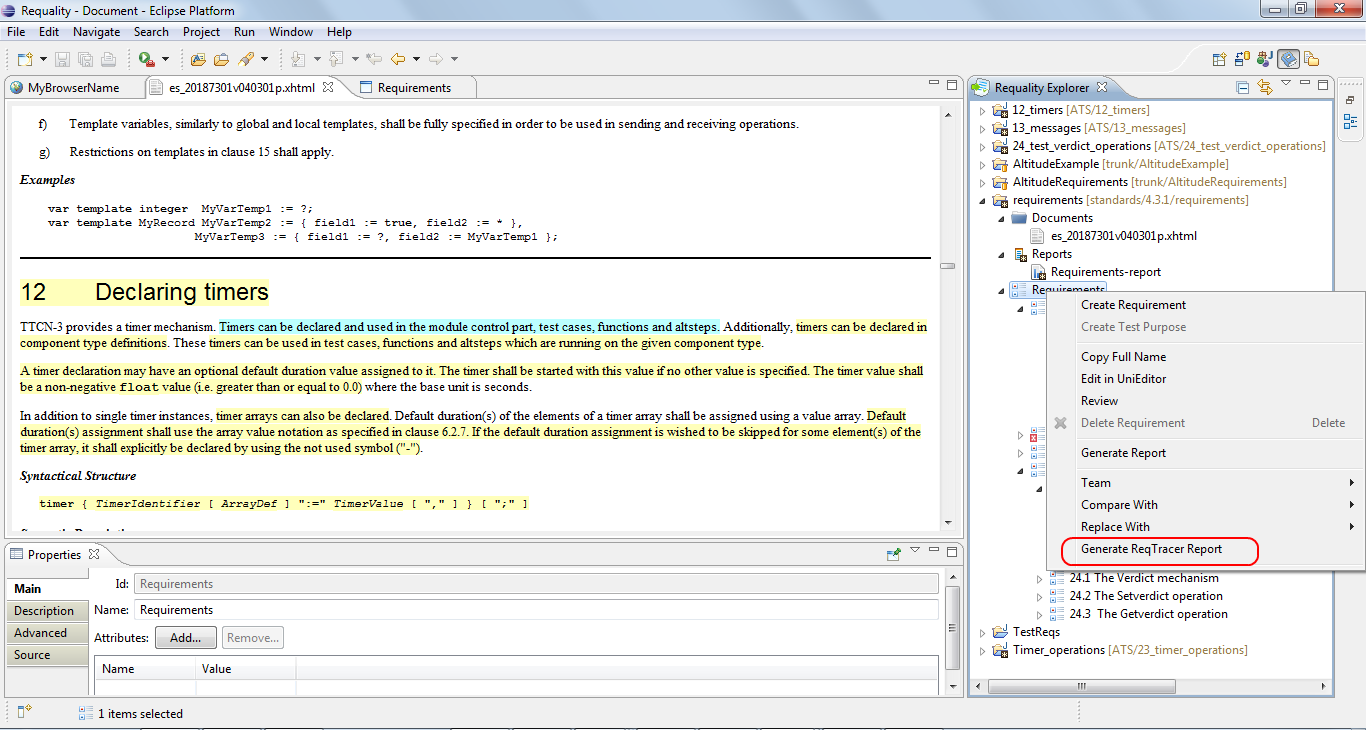


Figure 2 Invoking coverage report generator

In the dialog that pops up select the Eclipse projects that contain tests and enter the test extension. In STF 433 ATS it is *ttcn*.

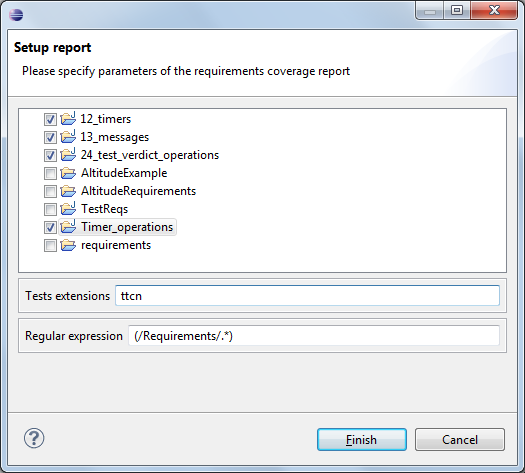


Figure 3 Configuring coverage report generator

The report is presented as a table with requirements coverage. To define the coverage we should introduce leaf requirements first. A requirement are considered to *leaf* is the requirement has no child requirements and no test purposes. A leaf requirement or a test purpose is said to be *covered* if there is a test case referring it. Coverage for non-leaf requirements is measured in percent. It is the number of covered leaf requirements and test purposes in the tree originating from the requirement divided by the total number of leaf requirements and test purposes in that tree:

Where Req is some requirement, CLF – number of covered leaf requirements in the Req tree, CTP – the number of covered test purposes, LF and TP – total number or leaf requirements and test purposes correspondingly in the tree originating from requirement Req.

This number is between 0 and 1 and is expressed as percentage of full coverage.

# Annex. Coverage report

**Report for "TTCN3 Specification" requirements collection**

| **Entity** | **Total** | **Covered** | **Uncovered** | **Partial** |
| --- | --- | --- | --- | --- |
| Requirements | 49 | 45 | 4 | 0 |
| Test purposes | 34 | 32 | 2 |  |

[Inconclusive requirements: 1](#inconcReqs)

|  |  |
| --- | --- |
| **Requirement: Requirements** | 92,5 |
| **Requirement: 12 Declaring timers**  12       Declaring timers | 100,0 |
| **Requirement: Timer can be declared in module control, test cases, functions, altsteps**  Timers can be declared and used in the module control part, test cases, functions and altsteps.  **Test cases:**   * [Sem\_12\_toplevel\_timer\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_002.ttcn) * [Sem\_12\_toplevel\_timer\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_003.ttcn) * [Sem\_12\_toplevel\_timer\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_004.ttcn) * [Sem\_12\_toplevel\_timer\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_005.ttcn) | 100,0 |
| **Requirement: Timers can be declared in component type**  timers can be declared in component type definitions  **Test cases:**   * [Sem\_12\_toplevel\_timer\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_001.ttcn) | 100,0 |
| **Requirement: Timers can be declared in component and used in test cases, functions, altsteps on this component**  timers can be used in test cases, functions and altsteps which are running on the given component type  **Test cases:**   * [NegSyn\_12\_toplevel\_timer\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_001.ttcn) * [Sem\_12\_toplevel\_timer\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_007.ttcn) * [Sem\_12\_toplevel\_timer\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_008.ttcn) * [Sem\_12\_toplevel\_timer\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_009.ttcn) | 100,0 |
| **Requirement: Timer value is non-neg float**  A timer declaration may have an optional default duration value assigned to it. The timer shall be started with this value if no other value is specified. The timer value shall be a non-negative **float** value (i.e. greater than or equal to 0.0)  **Test cases:**   * [NegSem\_12\_toplevel\_timer\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_001.ttcn) * [NegSem\_12\_toplevel\_timer\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_002.ttcn) * [Sem\_12\_toplevel\_timer\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Sem_12_toplevel_timer_006.ttcn) | 100,0 |
| **Requirement: timer array values are non-neg float or minus**  timer arrays can also be declared Default duration(s) assignment shall use the array value notation as specified in clause 6.2.7. If the default duration assignment is wished to be skipped for some element(s) of the timer array, it shall explicitly be declared by using the not used symbol ("-")  **Test cases:**   * [Syn\_12\_toplevel\_timer\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_004.ttcn) | 100,0 |
| **Requirement: Timer declaration syntax**  **timer** { *TimerIdentifier* [ *ArrayDef* ] ":=" *TimerValue* [ "," ] } [ ";" ]  **Test cases:**   * [NegSem\_12\_toplevel\_timer\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_003.ttcn) * [NegSem\_12\_toplevel\_timer\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_004.ttcn) * [NegSem\_12\_toplevel\_timer\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_005.ttcn) * [NegSem\_12\_toplevel\_timer\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_006.ttcn) * [NegSem\_12\_toplevel\_timer\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_007.ttcn) * [NegSem\_12\_toplevel\_timer\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSem_12_toplevel_timer_008.ttcn) * [NegSyn\_12\_toplevel\_timer\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_002.ttcn) * [NegSyn\_12\_toplevel\_timer\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_003.ttcn) * [NegSyn\_12\_toplevel\_timer\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_004.ttcn) * [NegSyn\_12\_toplevel\_timer\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_005.ttcn) * [NegSyn\_12\_toplevel\_timer\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_006.ttcn) * [NegSyn\_12\_toplevel\_timer\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/NegSyn_12_toplevel_timer_007.ttcn) * [Syn\_12\_toplevel\_timer\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_001.ttcn) * [Syn\_12\_toplevel\_timer\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_002.ttcn) * [Syn\_12\_toplevel\_timer\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_003.ttcn) * [Syn\_12\_toplevel\_timer\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_005.ttcn) * [Syn\_12\_toplevel\_timer\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/12_timers/12_toplevel/Syn_12_toplevel_timer_006.ttcn) | 100,0 |
| **Requirement: 13 Declaring messages** | 100,0 |
| **Requirement: Messages are instances of types declared in the in-out-inout clauses of message port type definition**  Messages are instances of types declared in the in/out/inout clauses of message port type definition.  **Test cases:**   * [Sem\_13\_declaring\_msg\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_001.ttcn) | 100,0 |
| **Requirement: Any type can be declared as type of message in a message port type**  Any type can be declared as type of a message in a message port type definition, i.e. values of any basic or structured type (see clauses 6.1 and 6.2) can be sent or received.  **Test cases:**   * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_001.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_002.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_003.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_004.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_005.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_006.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_007.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_008.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_009.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_010.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_011.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_012.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_013.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_013.ttcn) * [Sem\_13\_toplevel\_declaring\_msg\_various\_types\_014.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_toplevel_declaring_msg_various_types_014.ttcn) | 100,0 |
| **Requirement: Received messages can also be declared as a combination of value and matching mechanisms**  Received messages can also be declared as a combination of value and matching mechanisms  **Test cases:**   * [Sem\_13\_declaring\_msg\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_001.ttcn) * [Sem\_13\_declaring\_msg\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_002.ttcn) | 100,0 |
| **Requirement: Sorts of message insances**  Instances of messages can be declared by global, local or in-line templates (see clause 15) or being constructed and passed via variables or template variables parameters or template parameters | 100,0 |
| **Test purpose: Global template as a message**  -- A global template can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }         -- contents of T is not specified here         defining template T                  defining test case TC {             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending a global template             and invoking MTC.p.send(T)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_004.ttcn) | 100,0 |
| **Test purpose: Inline template as a message**  -- Inline template can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining test case TC {             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending an inline template             -- contents of template T is not specified here             and invoking MTC.p.send(inline template)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_006.ttcn) | 100,0 |
| **Test purpose: Local template as a message**  -- Local template can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining test case TC {             -- contents of template T is not specified here             defining template T             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending the local template             and invoking MTC.p.send(T)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_009.ttcn) | 100,0 |
| **Test purpose: Parameter as a message**  -- Parameter can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining function F(param) runnning on MTC {             -- Sending the value of the parameter             invoking MTC.p.send(param)         } where {             type of param is compatible with out type of MTC.p         }                  defining test case TC {             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending the local template             and invoking F(some value)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_007.ttcn) | 100,0 |
| **Test purpose: Template parameter as a message**  -- Template parameter can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining function F(template param) runnning on MTC {             -- Sending the template of the parameter             invoking MTC.p.send(param)         } where {             type of param is compatible with out type of MTC.p         }                  defining test case TC {             -- contents of template T is not specified here             defining template T             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending the local template as function parameter             and invoking F(T)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_008.ttcn) | 100,0 |
| **Test purpose: Template variable as a message**  -- Template variable can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining test case TC {             -- contents of template T is not specified here             defining template variable T\_var                 initialized by some value             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending the local template variable             and MTC.p.send(T\_var)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_005.ttcn) | 100,0 |
| **Test purpose: Variable as a message**  -- Variable can be sent and received in a message port  ensure that {     when {         IUT receives TTCN3 module         defining component PTC {             defining port p         }         defining component MTC {             defining port p         }                  defining test case TC {             -- contents of template T is not specified here             defining variable Var                 initialized by some value             creating component PTC             and connecting MTC.p with PTC.p             and starting component PTC                 with function {                     invoking alt statement {                         branch p.receive with setverdict(pass)                         branch timeout with setverdict(fail)                     }                 }             -- Sending the local variable             and MTC.p.send(Var)         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_13\_declaring\_msg\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_001.ttcn) * [Sem\_13\_declaring\_msg\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/13_messages/13_toplevel/Sem_13_declaring_msg_003.ttcn) | 100,0 |
| **Requirement: 23 Timer operations**  23       Timer operations The running-timers list and the timeout-list are only a conceptual lists and do not restrict the implementation of timers. Other data structures like a set, where the access to timeout events is not restricted by, e.g. the order in which the timeout events have happened, may also be used. | 96,2 |
| **Requirement: Usage of timers is allowed in test cases, functions, altsteps, module control**  TTCN‑3 supports a number of timer operations as given in table 27. These operations may be used in test cases, functions, altsteps and module control. | 100,0 |
| **Test purpose: Positive syntax test**  ensure that {     when {         IUT receives TTCN3 module         defining altstep {             declaring timer t             and containing t.start             and containing t.running             and containing t.read             and containing t.timeout             and containing t.stop         }         defining function {             declaring timer t             and containing t.start             and containing t.running             and containing t.read             and containing t.timeout             and containing t.stop         }         defining test case {             declaring timer t             and containing t.start             and containing t.running             and containing t.read             and containing t.timeout             and containing t.stop         }         defining module control {             declaring timer t             and containing t.start             and containing t.running             and containing t.read             and containing t.timeout             and containing t.stop         }     }     then {         IUT rejects the module     } }  **Test cases:**   * [Syn\_23\_toplevel\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/Syn_23_toplevel_001.ttcn) * [Syn\_23\_toplevel\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/Syn_23_toplevel_002.ttcn) | 100,0 |
| **Test purpose: Timer read is not allowed in component definitions**  when {      A test module contains timer operations in test cases, functions, altsteps and module control  }  ensure {     .The module compiles successfully }    **Test cases:**   * [NegSem\_23\_toplevel\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/NegSem_23_toplevel_001.ttcn) | 100,0 |
| **Test purpose: Timer running is not allowed in component definitions**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             defining timer t             defining boolean field                 initialized with t.running         }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSem\_23\_toplevel\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/NegSem_23_toplevel_002.ttcn) | 100,0 |
| **Test purpose: Timer stop is not allowed in module definitions**  ensure that {     when {         IUT receives TTCN3 module         containing "all timer.stop" in definitions part     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_23\_toplevel\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/NegSyn_23_toplevel_001.ttcn) | 100,0 |
| **Test purpose: Timer timeout operation is not allowed in module definitions**  ensure that {     when {         IUT receives TTCN3 module         containing "any timer.timeout" in definitions part     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_23\_toplevel\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/23_toplevel/NegSyn_23_toplevel_002.ttcn) | 100,0 |
| **Requirement: 23.2 The start timer operation**  23.2      The Start timer operation | 100,0 |
| **Requirement: Start timer syntax**  ( ( *TimerIdentifier* | *TimerParIdentifier* ) { "[" *SingleExpression* "]" } )  "." **start** [ "(" *TimerValue* ")" ]  **Test cases:**   * [NegSyn\_2302\_timer\_start\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_001.ttcn) * [NegSyn\_2302\_timer\_start\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_002.ttcn) * [NegSyn\_2302\_timer\_start\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_004.ttcn) * [NegSyn\_2302\_timer\_start\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_005.ttcn) * [NegSyn\_2302\_timer\_start\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_006.ttcn) * [NegSyn\_2302\_timer\_start\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_007.ttcn) * [NegSyn\_2302\_timer\_start\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_008.ttcn) * [NegSyn\_2302\_timer\_start\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_009.ttcn) * [NegSyn\_2302\_timer\_start\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_010.ttcn) * [NegSyn\_2302\_timer\_start\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_011.ttcn) * [NegSyn\_2302\_timer\_start\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_012.ttcn) * [NegSyn\_2302\_timer\_start\_013.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_013.ttcn) | 100,0 |
| **Requirement: Overridden timer value applies only to current instance**  The optional timer value parameter shall be used if no default duration is given, or if it is desired to override the default value specified in the timer declaration. When a timer duration is overridden, the new value applies only to the current instance of the timer, any later **start** operations for this timer, which do not specify a duration, shall use the default duration.  **Test cases:**   * [Sem\_2302\_timer\_start\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/Sem_2302_timer_start_003.ttcn) | 100,0 |
| **Requirement: neg timer value causes error verdict**  Starting a timer with a negative timer value, e.g. the timer value is the result of an expression, or without a specified timer value shall cause a runtime error.  **Test cases:**   * [NegSem\_2302\_timer\_start\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSem_2302_timer_start_003.ttcn) | 100,0 |
| **Requirement: timer can be restarted by repeated start operation**  The **start** operation may be applied to a running timer, in which case the timer is stopped and re-started. Any entry in a timeout-list for this timer shall be removed from the timeout-list.  **Test cases:**   * [Sem\_2302\_timer\_start\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/Sem_2302_timer_start_002.ttcn) | 100,0 |
| **Requirement: timer clock runs from 0 to the value set**  The timer clock runs from the float value zero (0.0) up to maximum stated by the duration parameter.  **Test cases:**   * [Sem\_2302\_timer\_start\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/Sem_2302_timer_start_001.ttcn) * [Sem\_2302\_timer\_start\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/Sem_2302_timer_start_003.ttcn) | 100,0 |
| **Requirement: Timer value is non-neg float**  Timer value shall be a non‑negative numerical **float** number (i.e. the value shall be greater or equal 0.0, infinity and not\_a\_number are disallowed).  **Test cases:**   * [NegSem\_2302\_timer\_start\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSem_2302_timer_start_001.ttcn) * [NegSem\_2302\_timer\_start\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSem_2302_timer_start_002.ttcn) * [NegSem\_2302\_timer\_start\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSem_2302_timer_start_003.ttcn) * [NegSem\_2302\_timer\_start\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSem_2302_timer_start_004.ttcn) | 100,0 |
| **Requirement: Timer with the timer value 0.0 times out immediately.**  Starting a timer with the timer value 0.0 means that the timer times out immediately.  **Test cases:**   * [Sem\_2302\_timer\_start\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/Sem_2302_timer_start_004.ttcn) | 100,0 |
| **Requirement: 23.3 The Stop timer operation**  23.3      The Stop timer operation | 100,0 |
| **Requirement: Timer stop syntax**  ( ( ( *TimerIdentifier* | *TimerParIdentifier* ) { "[" *SingleExpression* "]" } ) |  **all** **timer** )  "." **stop**  **Test cases:**   * [NegSyn\_2303\_timer\_stop\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_001.ttcn) * [NegSyn\_2303\_timer\_stop\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_002.ttcn) * [NegSyn\_2303\_timer\_stop\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_003.ttcn) * [NegSyn\_2303\_timer\_stop\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_004.ttcn) * [NegSyn\_2303\_timer\_stop\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_005.ttcn) * [NegSyn\_2303\_timer\_stop\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_006.ttcn) * [Syn\_2303\_timer\_stop\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Syn_2303_timer_stop_006.ttcn) * [Syn\_2303\_timer\_stop\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Syn_2303_timer_stop_007.ttcn) | 100,0 |
| **Requirement: stopped timer is inactive and elapsed time is 0**  A stopped timer becomes inactive and its elapsed time is set to the float value zero (0.0).  **Test cases:**   * [Sem\_2303\_timer\_stop\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Sem_2303_timer_stop_002.ttcn) | 100,0 |
| **Requirement: Stopping inactive timer is ok, but unobservable**  Stopping an inactive timer is a valid operation, although it does not have any effect. Stopping an expired timer causes the entry for this timer in the timeout-list to be removed.  **Test cases:**   * [Sem\_2303\_timer\_stop\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Sem_2303_timer_stop_004.ttcn) | 100,0 |
| **Requirement: all keyword can be used for timers in component or module control**  The **all** keyword may be used to stop all timers that have been started on a component or module control.  **Test cases:**   * [Sem\_2303\_timer\_stop\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Sem_2303_timer_stop_003.ttcn) | 100,0 |
| **Requirement: 23.4 The Read timer operation**  23.4      The Read timer operation | 100,0 |
| **Requirement: Timer Read syntax**  ( ( *TimerIdentifier* | *TimerParIdentifier* ) { "[" *SingleExpression* "]" } )  "." **read**  **Test cases:**   * [NegSyn\_2304\_timer\_read\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_001.ttcn) * [NegSyn\_2304\_timer\_read\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_002.ttcn) * [NegSyn\_2304\_timer\_read\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_003.ttcn) * [NegSyn\_2304\_timer\_read\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_004.ttcn) * [NegSyn\_2304\_timer\_read\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_005.ttcn) | 100,0 |
| **Requirement: Read returns elapsed time that is non-neg float**  The **read** operation returns the time that has elapsed since the specified timer was started. The returned value shall be of type **float**.  **Test cases:**   * [Sem\_2304\_timer\_read\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/Sem_2304_timer_read_002.ttcn) * [Sem\_2304\_timer\_read\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/Sem_2304_timer_read_003.ttcn) | 100,0 |
| **Requirement: Read on an inactive timer returns float zero**  Applying the **read** operation on an inactive timer, i.e. on a timer not listed on the running-timer list, will return the float value zero (0.0).  **Test cases:**   * [Sem\_2304\_timer\_read\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/Sem_2304_timer_read_001.ttcn) | 100,0 |
| **Requirement: 23.5 The Running timer operation**  23.5      The Running timer operation | 100,0 |
| **Requirement: Timer running syntax**  ( ( ( *TimerIdentifier* | *TimerParIdentifier* ) { "[" *SingleExpression* "]" } ) |  **any** **timer** )  "." **running**  **Test cases:**   * [NegSyn\_2305\_timer\_running\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_001.ttcn) * [NegSyn\_2305\_timer\_running\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_002.ttcn) * [NegSyn\_2305\_timer\_running\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_003.ttcn) * [NegSyn\_2305\_timer\_running\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_004.ttcn) * [NegSyn\_2305\_timer\_running\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_005.ttcn) * [NegSyn\_2305\_timer\_running\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_006.ttcn) * [Syn\_2306\_timer\_timeout\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Syn_2306_timer_timeout_001.ttcn) | 100,0 |
| **Requirement: The operation returns the value true if the timer is listed on the running list, false otherwise.**  The operation returns the value **true** if the timer is listed on the list, **false** otherwise.  **Test cases:**   * [Sem\_2305\_timer\_running\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/Sem_2305_timer_running_002.ttcn) * [Sem\_2305\_timer\_running\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/Sem_2305_timer_running_003.ttcn) * [Sem\_2305\_timer\_running\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/Sem_2305_timer_running_004.ttcn) | 100,0 |
| **Requirement: The any keyword may be used to check if any timer started on a component or module control is running**  The **any** keyword may be used to check if any timer started on a component or module control is running  **Test cases:**   * [Sem\_2305\_timer\_running\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/Sem_2305_timer_running_001.ttcn) | 100,0 |
| **Requirement: 23.6 The Timeout operation**  23.6      The Timeout operation | 100,0 |
| **Requirement: Timer Timeout syntax**  ( ( ( *TimerIdentifier* | *TimerParIdentifier* ) { "[" *SingleExpression* "]" } ) |  **any** **timer** )  "." **timeout**  **Test cases:**   * [NegSyn\_2306\_timer\_timeout\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_001.ttcn) * [NegSyn\_2306\_timer\_timeout\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_003.ttcn) * [NegSyn\_2306\_timer\_timeout\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_004.ttcn) * [NegSyn\_2306\_timer\_timeout\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_005.ttcn) * [NegSyn\_2306\_timer\_timeout\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_006.ttcn) * [NegSyn\_2306\_timer\_timeout\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_007.ttcn) | 100,0 |
| **Requirement: The any keyword used with the timeout operation succeeds if the timeout-list is not empty.**  The **any** keyword used with the **timeout** operation succeeds if the timeout-list is not empty. | 100,0 |
| **Test purpose: Any timer ignores timeouts in other components**  -- Create a test with a paralle test component. The PTC starts a timer, the MTC invokes an altstatement with any timer.timeout ensure that {     when {         IUT receives TTCN3 module         defining component PTC { empty }         and defining component MTC { empty }         and defining test case running on MTC {             creating PTC             and starting function on PTC {                 defining timers t1(time1) and t2(time2)                     indicating time1 < time2                 invoking t1.start                 invoking t2.start                 -- to ensure that PTC.done arrives later than potential t1.timeout                 invoking t2.timeout operation             }             and containig alt statement {                 -- timers from PTC must not be accessible from MTC                 branch 'any timer.timeout' with setverdict(fail)                 branch triggered by PTC.done with setverdict(pass)             }         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_005.ttcn) | 100,0 |
| **Test purpose: Wait for timers in scope of the alt**  -- Declare a few timers in a component. -- Define an altstep that waits for any timer and runs on that component. -- Start a few timers and invoke the altstep.  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             containing few timers         }         defining altstep  A() running on MTC {             with branch triggered by "any timer.timeout" with setverdict(pass)         }         defining test case {             running on MTC             and starting the timers of MTC             and invoking A()         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_004.ttcn) | 100,0 |
| **Test purpose: Wait for timers that are outside of scope**  -- Define an altstep with any timer.timeout alternative. Start a timer and invoke the altstep.  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining altstep  A() {             with branch triggered by "any timer.timeout" with setverdict(pass)         }         defining test case {             running on MTC             and defining timer t locally             and starting timer t             and invoking A()         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates in reasonable time             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_006.ttcn) | 100,0 |
| **Requirement: The timeout shall not be used in a boolean expression.**  The **timeout** shall not be used in a **boolean** expression.  **Test cases:**   * [NegSyn\_2306\_timer\_timeout\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_002.ttcn) | 100,0 |
| **Requirement: Timeout operational semantics**  When a **timeout** operation is processed, if a timer name is indicated, the timeout-list is searched according to the TTCN‑3 scope rules. If there is a timeout event matching the timer name, that event is removed from the timeout-list, and the **timeout** operation succeeds. | 100,0 |
| **Test purpose: Timeout a timer that is already timed out**  -- Start a timer. Wait for its timeout. Wait for its timeout once again (bound this wait with another timer to preven indefinite wait).  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1 locally             and defining timer t2 locally             and starting timer t1             and invoking t1.timeout             and starting t2             and containing alt statement {                 branch t1.timeout with setverdict(fail);                 branch t2.timeout with setverdict(pass);             }         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_002.ttcn) | 100,0 |
| **Test purpose: Timeout started timer**  -- Wait for timeout of several started timers with different maximum times.  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timers t1(time1), t2(time2), t3(time3)                 indicating time1 < time2 < time3             and starting timers t1, t2, t3             and containing alt statement {                 -- the timer with smallest time must timeout first                 branch t1.timeout with setverdict(pass);                 branch t2.timeout with setverdict(fail);                 branch t3.timeout with setverdict(fail);             }         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_003.ttcn) | 100,0 |
| **Test purpose: Timeout unstarted timer**  -- Wait for timeout of an unstarted timer. The waiting must be bounded with waiting of a started timer to prevent indefinite waiting.  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1(time1)             and defining timer t2(time2)                 indicating time1 < time2             and starting timer t2             and containing alt statement {                 -- unstarted timer must not timeout                 branch t1.timeout with setverdict(fail);                 branch t2.timeout with setverdict(pass);             }         }     }     then {         IUT accepts the module         and IUT executes the module         and IUT execution terminates             with verdict pass     } }  **Test cases:**   * [Sem\_2306\_timer\_timeout\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_001.ttcn) | 100,0 |
| **Requirement: The timeout can be used as alternative in alt**  The **timeout** can be used to determine an alternative in an **alt** statement  **Test cases:**   * [Sem\_2306\_timer\_timeout\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_003.ttcn) | 100,0 |
| **Requirement: The timeout can be used as a standalone statement**  The timeout can be used as stand-alone statement in a behaviour description. In the latter case a timeout operation is considered to be shorthand for an alt statement with the timeout operation as the only alternative  **Test cases:**   * [Sem\_2306\_timer\_timeout\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_007.ttcn) | 100,0 |
| **Requirement: 23.1 Timer mechanism**  Only one entry for any particular timer may appear in the timeout-list and running-timer list of the test component or module control for which the timer has been declared. 23.1      The timer mechanism | 0,0 |
| **Requirement: timer expires and becomes inactive**  When a timer expires, the timer becomes immediately inactive. | 0,0 |
| **Requirement: 23.7 Summary of use of any and all with timers**  23.7      Summary of use of any and all with timers  Table 28: Any and All with Timers   |  |  |  |  | | --- | --- | --- | --- | | Operation | Allowed | | Example | |  | any | all |  | | **start** |  |  |  | | **stop** |  | yes | **all timer.stop** | | **read** |  |  |  | | **running** | yes |  | **if (any timer.running) {…}** | | **timeout** | yes |  | **any timer.timeout** | | 100,0 |
| **Test purpose: Allow all timer.stop**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case TC running on MTC {             defining timer t1             and defining timer t2             and starting timer t1             and starting timer t2                      where { duration of t2 is longer than that of t1 }             and invoking { all timer.stop}             and setverdict(pass) if not t2.running and not t1.runnung             and setverdict(fail) otherwise         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_2303\_timer\_stop\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Sem_2303_timer_stop_003.ttcn) * [Syn\_2303\_timer\_stop\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/Syn_2303_timer_stop_007.ttcn) | 100,0 |
| **Test purpose: Allow any timer.running**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining float variable f\_var             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and setverdict(pass) if any timer.running produces true             and setverdict(fail) otherwise         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }    **Test cases:**   * [Sem\_2305\_timer\_running\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/Sem_2305_timer_running_004.ttcn) | 100,0 |
| **Test purpose: Allow any timer.timeout**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case TC running on MTC {             defining timer t1             and defining timer t2             and starting timer t1             and starting timer t2                      where { duration of t2 is longer than that of t1 }             and invoking { any timer.timeout}             and setverdict(pass) if t2.running and not t1.runnung             and setverdict(fail) otherwise         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }    **Test cases:**   * [Sem\_2306\_timer\_timeout\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_004.ttcn) * [Sem\_2306\_timer\_timeout\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_005.ttcn) * [Sem\_2306\_timer\_timeout\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_006.ttcn) | 100,0 |
| **Test purpose: Disallow all timer.read**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining float variable f\_var             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and assigning { f\_var := all timer.read }                          }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_2304\_timer\_read\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_005.ttcn) | 100,0 |
| **Test purpose: Disallow all timer.running**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining boolean variable b\_var             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and assigning { b\_var := all timer.running }                          }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_2305\_timer\_running\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2305_the_running_timer_operation/NegSyn_2305_timer_running_006.ttcn) | 100,0 |
| **Test purpose: Disallow all timer.start**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1             and defining timer t2             and invoking { all timer.start }         }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_2302\_timer\_start\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_012.ttcn) * [NegSyn\_2302\_timer\_start\_013.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_013.ttcn) | 100,0 |
| **Test purpose: Disallow all timer.timeout**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and invoking { all timer.timeout() }         }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_2306\_timer\_timeout\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/NegSyn_2306_timer_timeout_007.ttcn) | 100,0 |
| **Test purpose: Disallow any timer.read**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining float variable f\_var             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and assigning { f\_var := any timer.read }                          }     }     then {         IUT rejects the module     } }    **Test cases:**   * [NegSyn\_2304\_timer\_read\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2304_the_read_timer_operation/NegSyn_2304_timer_read_004.ttcn) | 100,0 |
| **Test purpose: Disallow any timer.start**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1             and defining timer t2             and invoking { any timer.start }         }     }     then {         IUT rejects the module     } }      **Test cases:**   * [NegSyn\_2302\_timer\_start\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_010.ttcn) * [NegSyn\_2302\_timer\_start\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2302_the_start_timer_operataion/NegSyn_2302_timer_start_011.ttcn) | 100,0 |
| **Test purpose: Disallow any timer.stop**  ensure that {     when {         IUT receives TTCN3 module         defining component MTC {             empty         }         defining test case running on MTC {             defining timer t1             and defining timer t2             and starting timer t1             and starting t2             and invoking { any timer.stop() }         }     }     then {         IUT rejects the module     } }  **Test cases:**   * [NegSyn\_2303\_timer\_stop\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2303_the_stop_timer_operation/NegSyn_2303_timer_stop_006.ttcn) | 100,0 |
| **Requirement: 24 Test verdict operations** | 76,2 |
| **Requirement: Getverdict and setverdict operations shall only be used in test cases, altsteps and functions.**  Verdict operations given in table 29 allow to set and retrieve verdicts. These operations shall only be used in test cases, altsteps and functions  **Test cases:**   * [NegSem\_24\_toplevel\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/NegSem_24_toplevel_001.ttcn) * [NegSem\_24\_toplevel\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/NegSem_24_toplevel_002.ttcn) * [NegSem\_24\_toplevel\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/NegSem_24_toplevel_003.ttcn) * [NegSem\_24\_toplevel\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/NegSem_24_toplevel_004.ttcn) * [NegSem\_24\_toplevel\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/NegSem_24_toplevel_005.ttcn) | 100,0 |
| **Requirement: Setverdict locations**  Setverdict operations shall only be used in test cases, altsteps and functions. | 100,0 |
| **Test purpose: Setverdict allowed in altsteps**  ensure that {     when {         IUT receives TTCN3 module         defining altstep A() {             branch with arbitrary trigger: setverdict(pass)         }     }     then {         IUT accepts the module     } }  **Test cases:**   * [Syn\_24\_toplevel\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/Syn_24_toplevel_003.ttcn) | 100,0 |
| **Test purpose: Setverdict allowed in test cases**  ensure that {     when {         IUT receives TTCN3 module         defining test case TC {             setverdict(pass)         }     }     then {         IUT accepts the module     } }  **Test cases:**   * [Syn\_24\_toplevel\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/Syn_24_toplevel_002.ttcn) | 100,0 |
| **Test purpose: Setverdict in functions allowed**  ensure that {     when {         IUT receives TTCN3 module         defining function F {             setverdict(pass)         }     }     then {         IUT accepts the module     } }  **Test cases:**   * [Syn\_24\_toplevel\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/24_toplevel/Syn_24_toplevel_001.ttcn) | 100,0 |
| **Requirement: 24.1 The Verdict mechanism** | 50,0 |
| **Requirement: Global test case verdict**  there is a global test case verdict instantiated and handled by the test system that is updated when each test component (i.e. the MTC and each and every PTC) terminates execution (see figure 14). This verdict is not accessible to the **getverdict** and **setverdict** operations. The value of this verdict shall be returned by the test case when it terminates execution. | 0,0 |
| **Test purpose: Ensure getverdict returns local verdict** | 0,0 |
| **Test purpose: Ensure setverdict modifies only local verdict** | 0,0 |
| **Requirement: Five different values of verdict**  The verdict can have five different values: **pass**, **fail**, **inconc**, **none** and **error**, i.e. the distinguished values of the **verdicttype** (see clause 6.1)  **Test cases:**   * [Syn\_2401\_FiveValues\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Syn_2401_FiveValues_001.ttcn) | 100,0 |
| **Requirement: Local verdict is initialized with none**  When a test component is instantiated, its local verdict object is created and set to the value **none**.  **Test cases:**   * [Sem\_2401\_InitiallyNone\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_InitiallyNone_001.ttcn) | 100,0 |
| **Requirement: Overwriting rules for setverdict**  When changing the value of the local verdict (i.e. using the **setverdict** operation) the effect of this change shall follow the overwriting rules listed in table 30.  **Test cases:**   * [Sem\_2401\_GlobalVerdict\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_001.ttcn) * [Sem\_2401\_GlobalVerdict\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_002.ttcn) * [Sem\_2401\_GlobalVerdict\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_003.ttcn) * [Sem\_2401\_GlobalVerdict\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_004.ttcn) * [Sem\_2401\_GlobalVerdict\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_005.ttcn) * [Sem\_2401\_GlobalVerdict\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_006.ttcn) * [Sem\_2401\_GlobalVerdict\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_007.ttcn) * [Sem\_2401\_GlobalVerdict\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_008.ttcn) * [Sem\_2401\_GlobalVerdict\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_009.ttcn) * [Sem\_2401\_GlobalVerdict\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_010.ttcn) * [Sem\_2401\_GlobalVerdict\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_011.ttcn) * [Sem\_2401\_GlobalVerdict\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_GlobalVerdict_012.ttcn) * [Sem\_2401\_LocalVerdict\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_001.ttcn) * [Sem\_2401\_LocalVerdict\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_002.ttcn) * [Sem\_2401\_LocalVerdict\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_003.ttcn) * [Sem\_2401\_LocalVerdict\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_004.ttcn) * [Sem\_2401\_LocalVerdict\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_005.ttcn) * [Sem\_2401\_LocalVerdict\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_006.ttcn) * [Sem\_2401\_LocalVerdict\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_007.ttcn) * [Sem\_2401\_LocalVerdict\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_008.ttcn) * [Sem\_2401\_LocalVerdict\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_009.ttcn) * [Sem\_2401\_LocalVerdict\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_010.ttcn) * [Sem\_2401\_LocalVerdict\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_011.ttcn) * [Sem\_2401\_LocalVerdict\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_012.ttcn) | 100,0 |
| **Requirement: Overwriting rules for global verdict**  The test case verdict is implicitly updated on the termination of a test component. The effect of this implicit operation shall also follow the overwriting rules listed in table 30. | 0,0 |
| **Requirement: Error verdict shall not be set with setverdict**  The **error** verdict is special in that it is set by the test system to indicate that a test case (i.e. run-time) error has occurred. It shall not be set by the **setverdict** operation  **Test cases:**   * [NegSem\_2401\_SetverdictError.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/NegSem_2401_SetverdictError.ttcn) | 100,0 |
| **Requirement: 24.2 The Setverdict operation** | 80,0 |
| **Requirement: The value of the local verdict is changed with the setverdict operation.**  The value of the local verdict is changed with the **setverdict** operation. The effect of this change shall follow the overwriting rules listed in table 30  **Test cases:**   * [Sem\_2401\_LocalVerdict\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_001.ttcn) * [Sem\_2401\_LocalVerdict\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_002.ttcn) * [Sem\_2401\_LocalVerdict\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_003.ttcn) * [Sem\_2401\_LocalVerdict\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_004.ttcn) * [Sem\_2401\_LocalVerdict\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_005.ttcn) * [Sem\_2401\_LocalVerdict\_006.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_006.ttcn) * [Sem\_2401\_LocalVerdict\_007.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_007.ttcn) * [Sem\_2401\_LocalVerdict\_008.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_008.ttcn) * [Sem\_2401\_LocalVerdict\_009.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_009.ttcn) * [Sem\_2401\_LocalVerdict\_010.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_010.ttcn) * [Sem\_2401\_LocalVerdict\_011.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_011.ttcn) * [Sem\_2401\_LocalVerdict\_012.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2401_the_verdict_mechanism/Sem_2401_LocalVerdict_012.ttcn) | 100,0 |
| **Requirement: Optional setverdict parameters**  The optional parameters allow to provide information that explain the reasons for assigning the verdict. This information is composed to a string and stored in an implicit **charstring** variable. On termination of the test component, the actual local verdict is logged together with the implicit **charstring** variable.  **Test cases:**   * [Sem\_2402\_setverdict\_params\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/Sem_2402_setverdict_params_001.ttcn) * [Sem\_2402\_setverdict\_params\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/Sem_2402_setverdict_params_002.ttcn) * [Sem\_2402\_setverdict\_params\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/Sem_2402_setverdict_params_003.ttcn) | 100,0 |
| **Requirement: Setverdict allowed verdict values**  The **setverdict** operation shall only be used with the values **pass**, **fail**, **inconc** and **none**. It shall not be used to assign the value **error**, this is set by the test system only to indicate run-time errors.  **Test cases:**   * [NegSem\_2402\_setverdict\_params\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/NegSem_2402_setverdict_params_001.ttcn) * [NegSem\_2402\_setverdict\_params\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/NegSem_2402_setverdict_params_002.ttcn) * [NegSem\_2402\_setverdict\_params\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/NegSem_2402_setverdict_params_003.ttcn) * [NegSem\_2402\_setverdict\_params\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/NegSem_2402_setverdict_params_004.ttcn) * [NegSem\_2402\_setverdict\_params\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/NegSem_2402_setverdict_params_005.ttcn) | 100,0 |
| **Requirement: SingleExpression shall resolve to a value of type verdict**  *SingleExpression* shall resolve to a value of type **verdict** | 0,0 |
| **Requirement: For FreeText and TemplateInstance, the same rules and restrictions apply as for the parameters of the log statement.**  For *FreeText* and *TemplateInstance*,the same rules and restrictions apply as forthe parameters of the **log** statement.  **Test cases:**   * [Sem\_2402\_setverdict\_logging\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2402_the_setverdict_operation/Sem_2402_setverdict_logging_001.ttcn) | 100,0 |
| **Requirement: 24.3 The Getverdict operation** | 100,0 |
| **Requirement: Getverdict returns local verdict value**  The **getverdict** operation returns the actual value of the local verdict.  **Test cases:**   * [Sem\_2403\_getverdict\_001.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2403_the_getverdict_operation/Sem_2403_getverdict_001.ttcn) * [Sem\_2403\_getverdict\_002.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2403_the_getverdict_operation/Sem_2403_getverdict_002.ttcn) * [Sem\_2403\_getverdict\_003.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2403_the_getverdict_operation/Sem_2403_getverdict_003.ttcn) * [Sem\_2403\_getverdict\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2403_the_getverdict_operation/Sem_2403_getverdict_004.ttcn) | 100,0 |
| **Test purpose: Getverdict returns none for uninitialized verdict**  -- When test case does not invoke setverdict() directly or indirectly getverdict returns 'none'  ensure that {     when {         IUT receives TTCN3 module         defining test case TC {             if getverdict returns none                 setverdict pass             else                 setverdict fail         }     }     then {         IUT accepts the module         and IUT executes TC             with verdict pass     } }  **Test cases:**   * [Sem\_2403\_getverdict\_005.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/24_test_verdict_operations/2403_the_getverdict_operation/Sem_2403_getverdict_005.ttcn) | 100,0 |

* **Inconclusive requirements**
* **/Requirements/23 Timer operations/23.6 The Timeout operation//Requirements/23 Timer operations/23.6 The Timeout operation/The any keyword used with the timeout operation succeeds if the timeout-list is not empty./Wait for timers in scope of the alt**

Mentioned in files:

* [Sem\_2306\_timer\_timeout\_004.ttcn](file://C:/work/ETSI-TTCN3/STF%20433/ttwb-workspace/Timer_operations/2306_the_timeout_operation/Sem_2306_timer_timeout_004.ttcn)

1. ES 201 873-1 TTCN-3: Core Language, v. 4.3.1 [↑](#footnote-ref-1)
2. http://requality.org/, in Russian at the moment of writing. [↑](#footnote-ref-2)
3. Final draft ES 202 553 V2.0.0 (2012-02). **Methods for Testing and Specification (MTS);** Extensible notation for expressing Test Purposes, Requirements and Assertions (ExTRA) [↑](#footnote-ref-3)