


# Statement of compliance for Eclipse Titan

Adrien Kirjak

Version 11.2.1, 2026-08-05

# Table of Contents


1. Description . . . 2
2. References . . . 3
   2.1. Normative references . . . 3
   2.2. Informative references . . . 3
3. Definitions and abbreviations . . . 4
   3.1. Definitions . . . 4
   3.2. Abbreviations . . . 4
4. Instructions for completing the ICS proforma . . . 6
   4.1. Other information . . . 6
   4.2. Identification of the implementation . . . 8
5. ICS proforma tables . . . 9
   5.1. Global statement of conformance . . . 9
   5.2. Basic language elements . . . 9
   5.3. Identifiers and keywords . . . 9
   5.4. Scope rules . . . 9
   5.5. Scope of formal parameters . . . 10
   5.6. Uniqueness of identifiers . . . 10
   5.7. Ordering of language elements . . . 12
   5.8. Parameterization . . . 12
   5.9. Formal parameters . . . 13
   5.10. Formal parameters of kind value . . . 16
   5.11. Formal parameters of kind template . . . 22
   5.12. Formal parameters of kind timer . . . 28
   5.13. Formal parameters of kind port . . . 29
   5.14. Actual parameters . . . 30
   5.15. Cyclic definitions . . . 72
   5.16. Simple basic types and values . . . 73
   5.17. Basic string types and values . . . 75
   5.18. Accessing individual string elements . . . 77
   5.19. Lists of values . . . 78
   5.20. Lists of types . . . 79
   5.21. Ranges . . . 80
   5.22. String length restrictions . . . 82
   5.23. Pattern subtyping of character string types . . . 84
   5.24. Mixing patterns, lists and ranges . . . 85
   5.25. Using length restriction with other constraints . . . 85
   5.26. Structured types and values . . . 86
   5.27. Record type and values . . . 88

5.28. Referencing fields of a record type . . . 89
5.29. Set type and values . . . 92
5.30. Records and sets of single types . . . 93
5.31. Referencing elements of record of and set of types . . . 96
5.32. Enumerated type and values . . . 99
5.33. Unions . . . 102
5.34. Referencing fields of a union type . . . 103
5.35. Option and union . . . 105
5.36. Anytype . . . 105
5.37. Arrays . . . 107
5.38. The default type . . . 112
5.39. Communication port types . . . 112
5.40. Component types . . . 114
5.41. Addressing entities inside the SUT . . . 115
5.42. Subtyping of structured types . . . 116
5.43. Type compatibility of non-structured types . . . 117
5.44. Type compatibility of structured types . . . 119
5.45. Type compatibility of enumerated types . . . 121
5.46. Type compatibility of component types . . . 122
5.47. Type compatibility of communication operations . . . 123
5.48. Expression . . . 124
5.49. Arithmetic operators . . . 125
5.50. List operator . . . 131
5.51. Relational operators . . . 132
5.52. Logical operators . . . 137
5.53. Bitwise operators . . . 137
5.54. Shift operators . . . 138
5.55. Rotate operators . . . 138
5.56. Field references and list elements . . . 138
5.57. Definition of a module . . . 139
5.58. Module definitions part . . . 140
5.59. Module parameters . . . 140
5.60. Groups of definitions . . . 141
5.61. General format of import . . . 142
5.62. Importing single definitions . . . 148
5.63. Importing groups . . . 148
5.64. Importing definitions of the same kind . . . 149
5.65. Importing all definitions of a module . . . 150
5.66. Import definitions from other TTCN-3 editions and from non-TTCN-3 modules . . . 150
5.67. Importing of `import` statements from TTCN-3 modules . . . 151
5.68. Compatibility of language specifications of imports . . . 152

5.69. Definition of friend modules . . . 153
5.70. Visibility of definitions . . . 153
5.71. Module control part . . . 154
5.72. Port types, component types and test configurations . . . 155
5.73. Communication ports . . . 156
5.74. Declaring constants . . . 158
5.75. Value variables . . . 160
5.76. Template variables . . . 160
5.77. Declaring timers . . . 161
5.78. Declaring messages . . . 164
5.79. Declaring procedure signatures . . . 167
5.80. Declaring templates . . . 168
5.81. Declaring message templates . . . 168
5.82. Declaring signature templates . . . 169
5.83. Global and local templates . . . 170
5.84. In-line templates . . . 172
5.85. Modified templates . . . 172
5.86. Referencing individual string elements . . . 175
5.87. Referencing record and set fields . . . 175
5.88. Referencing record of and set of elements . . . 176
5.89. Referencing signature parameters . . . 179
5.90. Referencing union alternatives . . . 180
5.91. Template restrictions . . . 181
5.92. `Match` operation . . . 189
5.93. `Valueof` operation . . . 191
5.94. Concatenating templates of string and list types . . . 192
5.95. Functions . . . 195
5.96. Invoking functions . . . 196
5.97. Predefined functions . . . 196
5.98. External functions . . . 209
5.99. Invoking function from specific places . . . 210
5.100. Altsteps . . . 248
5.101. Invoking altsteps . . . 249
5.102. Test cases . . . 250
5.103. Assignments . . . 250
5.104. The `if-else` statement . . . 251
5.105. The `Select` statements . . . 252
5.106. The `select union` statement . . . 252
5.107. The `for` statement . . . 254
5.108. The `while` statement . . . 254
5.109. The `do-while` statement . . . 255

5.110. The `label` statement . . . 255
5.111. The `goto` statement . . . 256
5.112. The `stop` execution statement . . . 256
5.113. The `return` statement . . . 257
5.114. The `log` statement . . . 257
5.115. The `continue` statement . . . 258
5.116. Statement and operations for alternative behaviours . . . 258
5.117. The `alt` statement . . . 258
5.118. The `repeat` statement . . . 269
5.119. The `interleave` statement . . . 270
5.120. The default mechanism . . . 271
5.121. The `activate` operation . . . 273
5.122. `Connection` operations . . . 274
5.123. The `connect` and `map` operations . . . 275
5.124. The `disconnect` and `unmap` operations . . . 278
5.125. Test case operations . . . 281
5.126. The `create` operation . . . 281
5.127. The `start test` component operation . . . 282
5.128. The `stop` test behaviour operation . . . 284
5.129. The `kill` test component operation . . . 286
5.130. The `alive` operation . . . 287
5.131. The `running` operation . . . 289
5.132. The `done` operation . . . 292
5.133. The `killed` operation . . . 294
5.134. The `send` operation . . . 297
5.135. The `receive` operation . . . 299
5.136. The `trigger` operation . . . 303
5.137. The `call` operation . . . 307
5.138. The `getcall` operation . . . 311
5.139. The `reply` operation . . . 315
5.140. The `getreply` operation . . . 316
5.141. The `raise` operation . . . 321
5.142. The `catch` operation . . . 322
5.143. The `check` operation . . . 325
5.144. `Timer` operations . . . 339
5.145. The `start` timer operation . . . 339
5.146. The `stop` timer operation . . . 341
5.147. The `read` timer operation . . . 342
5.148. The `running` timer operation . . . 343
5.149. The `timeout` operation . . . 344
5.150. Test `verdict` operations . . . 345

5.151. The verdict mechanism . . . 346
5.152. The setverdict mechanism . . . 348
5.153. The getverdict mechanism . . . 349
5.154. Module control . . . 350
5.155. The `execute` statement . . . 351
5.156. The control part . . . 352
5.157. Scope of attributes . . . 355
5.158. Optional attributes . . . 356
5.159. Matching specific values . . . 357
5.160. Value list . . . 358
5.161. Complemented value list . . . 359
5.162. Any value . . . 360
5.163. Any value or none . . . 361
5.164. Value range . . . 361
5.165. SuperSet . . . 362
5.166. SubSet . . . 364
5.167. Omitting optional fields . . . 365
5.168. Decoded content . . . 366
5.169. Enumerated value list . . . 366
5.170. Any element . . . 366
5.171. Any number of elements of no element . . . 367
5.172. Permutation . . . 368
5.173. Length restrictions . . . 369
5.174. The `ifpresent` indicator . . . 370
5.175. Matching character pattern . . . 370
5.176. Set expression . . . 370
5.177. Reference expression . . . 371
5.178. Match expression n times . . . 373
5.179. Match a referenced character set . . . 373
5.180. Type compatibility rules for patterns . . . 374
5.181. Case insensitive pattern matching . . . 375
5.182. Other functions . . . 375
5.183. Preprocessing macros . . . 375

## Abstract

The present document provides the Implementation Conformance Statement (ICS) proforma for the conformance test suite for the Eclipse Titan TTCN-3 implementation.

### Copyright

Copyright (c) 2000-2026 Ericsson Telecom AB
All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v2.0 that accompanies this distribution, and is available at
https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.html

### Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

# Chapter 1. Description

The present document provides the Implementation Conformance Statement (ICS) proforma for the conformance test suite for the Eclipse Titan TTCN-3 implementation as defined in ES 201 873-1 in compliance with the relevant guidance given in the proforma for TTCN-3 reference test suite TS 102 995.

In the present document only the core language features, specified in ES 201 873-1 have been considered but not

- the tool implementation (see Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI)
  and ETSI ES 201 873-6: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI).),
- language mapping (see Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 7: Using ASN.1 with TTCN-3,
  Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 8: The IDL to TTCN-3 Mapping and Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3)
- and language extension (see e.g. Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Configuration and Deployment Support. , Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization and Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types) aspects.

# Chapter 2. References

## 2.1. Normative references

The following referenced documents are necessary for the application of the present document.

1. ETSI ES 201 873-1: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language v4.7.1.
2. ISO/IEC 9646-7 (1994): Conformance testing methodology and framework - Part 7: Implementation Conformance Statement.
3. ISO/IEC 9646-1 (1992): Information Technology - Open Systems Interconnection - Conformance Testing Methodology and Framework - Part 1: General concepts.
4. ETSI TS 102 995: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Proforma for TTCN-3 reference test suite

## 2.2. Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

5. ETSI ES 201 873-5: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI).
6. ETSI ES 201 873-6: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI).
7. ETSI ES 201 873-7: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 7: Using ASN.1 with TTCN-3.
8. ETSI ES 201 873-8: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 8: The IDL to TTCN-3 Mapping.
9. ETSI ES 201 873-9: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3.
10. ETSI ES 202 781: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Configuration and Deployment Support.
11. ETSI ES 202 784: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization.
12. ETSI ES 202 785: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types.

# Chapter 3. Definitions and abbreviations

## 3.1. Definitions

**Abstract Test Suite (ATS):**
Test suite composed of abstract test cases

**Implementation Conformance Statement (ICS):**
Statement made by the supplier of an implementation claimed to conform to a given specification, stating which capabilities have been implemented

**ICS proforma:**
Document, in the form of a questionnaire, which when completed for an implementation or system becomes an ICS

**Implementation eXtra Information for Testing (IXIT):**
Statement made by a supplier or implementor of an IUT which contains or references all of the information related to the IUT and its testing environment, which will enable the test laboratory to run an appropriate test suite against the IUT

**IXIT proforma:**
Document, in the form of a questionnaire, which when completed for the IUT becomes the IXIT

**Implementation Under Test (IUT):**
Implementation of one or more OSI protocols in an adjacent user/provider relationship, being part of a real open system which is to be studied by testing

## 3.2. Abbreviations

**ATS**

Abstract Test Suite

**BNF**

Backus Naur Form

**ICS**

Implementation Conformance Statement

**IUT**

Implementation under Test

**IXIT**

Implementation eXtra Information for Testing

**SUT**

System Under Test

**TC**

Test Case

**TCI**

TTCN-3 Control Interface

**TP**

Test Purpose

**TRI**

TTCN-3 Runtime Interface

**TS**

Test System

**TSS**

Test Suite Structure

**TSS&TP**

Test Suite Structure and Test Purposes

**TTCN-3**

Testing and Test Control Notation edition 3

# Chapter 4. Instructions for completing the ICS proforma

## 4.1. Other information

More detailed instructions are given at the beginning of the different clauses of the ICS proforma.

The supplier of the implementation shall complete the ICS proforma in each of the spaces provided. If necessary, the supplier may provide additional comments separately in Clause A.4.

### 4.1.1. Purposes and structure

The purpose of this ICS proforma is to provide a mechanism whereby a TTCN-3 tool vendor of the TTCN-3 core language may provide information about the implementation in a standardized manner.

The ICS proforma is subdivided into clauses for the following categories of information:

- instructions for completing the ICS proforma;
- identification of the implementation;
- ICS proforma tables (containing the global statement of conformance).

### 4.1.2. Conventions

The ICS proforma is composed of information in tabular form in accordance with the guidelines presented in ISO/IEC 96467 .

- Item column

It contains a number that identifies the item in the table.

- Item description column

It describes each respective item (e.g. parameters, timers, etc.).

- Reference column

It gives reference to the TTCN-3 core language, except where explicitly stated otherwise.

- Status column

The following notations, defined in ISO/IEC 96467 , are used for the status column:

**m**
: mandatory - the capability is required to be supported.

**n/a**
: not applicable - in the given context, it is impossible to use the capability. No answer in the

support column is required.

**u**

undecided

**o**

optional - the capability may be supported or not.

**o.i**

qualified optional - for mutually exclusive or selectable options from a set. `i` is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table.

**ci**

conditional - the requirement on the capability ("m", "o" or "n/a") depends on the support of other optional or conditional items. `i` is an integer identifying a unique conditional status expression that is defined immediately following the table. For nested conditional expressions, the syntax `IF ⋯ THEN (IF ⋯ THEN ⋯ ELSE⋯) ELSE ⋯` shall be used to avoid ambiguities. If an `ELSE` clause is omitted, `ELSE n/a` shall be implied.

| NOTE | Support of a capability means that the capability is implemented in conformance to the TTCN-3 core language. |
|---|---|

- Support column

The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 96467 [2], are used for the support column:

Y or y supported by the implementation.

N or n not supported by the implementation.

N/A or n/a or "no answer required" (allowed only if the status is N/A, directly or after evaluation of a conditional status).

- Values allowed column

This column contains the values or the ranges of values allowed.

- Values supported column

The support column shall be filled in by the supplier of the implementation. In this column the values or the ranges of values supported by the implementation shall be indicated.

- References to items

For each possible item answer (answer in the support column) within the ICS proforma, a unique reference exists. It is defined as the table identifier, followed by a slash character "/", followed by the item number in the table. If there is more than one support column in a table, the columns shall be discriminated by letters (a, b, etc.) respectively.

EXAMPLE: 5/4 is the reference to the answer of item 4 in Table 5.

## 4.2. Identification of the implementation

Identification of the Implementation under Test (IUT) and the system in which it resides - the System Under Test (SUT) should be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information should both be filled in if they are different.

A person who can answer queries regarding information supplied in the ICS should be named as the contact person.

### 4.2.1. Date of the statement

| **Date of the statement:** | **2016.05.09** |
|---|---|

### 4.2.2. Implementation under Test (IUT) identification

| **IUT name:** | **Eclipse Titan** |
|---|---|
| IUT version: | CRL 113 200/5 R5A |

### 4.2.3. ICS contact person

| **Name:** | **Elemer Lelik** |
|---|---|
| Telephone number: | |
| Facsimile number: | |
| E-mail address: | Elemer.Lelik@ericsson.com |
| Additional information: | |

# Chapter 5. ICS proforma tables

## 5.1. Global statement of conformance

| | (Yes/No) |
|---|---|
| Are all mandatory capabilities implemented? | |

**NOTE** Nonsupported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is nonconforming.

## 5.2. Basic language elements

*Table 1. Basic language elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_05_TopLevel_001 | When the IUT loads a module containing some definitions before the module declaration then the module is rejected. | Clause 5 | m | y |

## 5.3. Identifiers and keywords

*Table 2. Identifiers and keywords*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0501_Identifier_001 | Cannot pass a charstring value to an integer variable. | Clause 5.1 | m | y |
| 2 | NegSyn_0501_Identifier_001 | When the IUT loads a module containing an identifier named with a keyword then the module is rejected. | Clause 5.1 | m | y |
| 3 | Syn_0501_Identifier_001 | The IUT handle the identifiers case sensitively. | Clause 5.1 | m | y |

## 5.4. Scope rules

*Table 3. Scope rules*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0502_Scope_001 | The IUT correctly handles definitions of local scope | Clause 5.2 | m | y |
| 2 | NegSem_0502_Scope_002 | The IUT correctly handles definitions of local scope | Clause 5.2 | m | y |
| 3 | NegSem_0502_Scope_003 | The IUT correctly handles definitions of local scope | Clause 5.2 | m | y |
| 4 | Sem_0502_Scope_001 | The IUT handle scope hieararchy of component constants. | Clause 5.2 | m | y |
| 5 | Sem_0502_Scope_002 | The IUT handle scope hieararchy with component booleans. | Clause 5.2 | m | y |
| 6 | Sem_0502_Scope_003 | The IUT handles scope hierarchy via functions. | Clause 5.2 | m | y |
| 7 | Sem_0502_Scope_004 | The IUT correctly handles the scope of definitions made in the module part. | Clause 5.2 | m | y |
| 8 | Sem_0502_Scope_008 | The IUT correctly handles definitions of extended component scope | Clause 5.2 | m | y |
| 9 | Syn_0502_Scope_001 | The IUT supports all the nine scope units. | Clause 5.2 | m | y |

## 5.5. Scope of formal parameters

*Table 4. Scope of formal parameters*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_050201_Scope_of_parameters_001 | The IUT correctly handles scope of formal function parameters | Clause 5.2.1 | m | y |
| 2 | Sem_050201_Scope_of_parameters_002 | The IUT correctly handles scope of formal function parameters | Clause 5.2.1 | m | y |

## 5.6. Uniqueness of identifiers

*Table 5. Uniqueness of identifiers*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_050202_Uniqueness_001 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 2 | NegSem_050202_Uniqueness_004 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 3 | NegSem_050202_Uniqueness_005 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 4 | NegSem_050202_Uniqueness_006 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 5 | NegSem_050202_Uniqueness_007 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 6 | NegSem_050202_Uniqueness_008 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 7 | NegSem_050202_Uniqueness_009 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 8 | NegSem_050202_Uniqueness_010 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 9 | NegSem_050202_Uniqueness_011 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | n |
| 10 | NegSem_050202_Uniqueness_012 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | n |
| 11 | Sem_050202_Uniqueness_001 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 12 | Sem_050202_Uniqueness_002 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |
| 13 | Sem_050202_Uniqueness_003 | The IUT correctly handles the uniqueness of variable names in its scope | Clause 5.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_050202_Uniqueness_004 | Identifiers for fields of structured types, enumerated values and groups do not have to be globally unique | Clause 5.2.2 | m | y |
| 15 | Sem_050202_Uniqueness_005 | Identifiers for fields of structured types, enumerated values and groups do not have to be globally unique | Clause 5.2.2 | m | y |

## 5.7. Ordering of language elements

*Table 6. Ordering of language elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0503_Ordering_001 | Declarations are in the allowed ordering | Clause 5.3 | m | y |
| 2 | NegSem_0503_Ordering_002 | Declarations are in the allowed ordering | Clause 5.3 | m | n |
| 3 | NegSem_0503_Ordering_003 | Declarations are in the allowed ordering | Clause 5.3 | m | n |
| 4 | Sem_0503_Ordering_001 | Allowed orderings of declarations are supported | Clause 5.3 | m | y |
| 5 | Sem_0503_Ordering_002 | Allowed any ordering with component definitions are supported | Clause 5.3 | m | y |
| 6 | Sem_0503_Ordering_005 | Allowed orderings of declarations are supported | Clause 5.3 | m | y |

## 5.8. Parameterization

*Table 7. Parameterization*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0504_parametrization_incompatibility_001 | The IUT correctly handles received testcase parametrization type incompatibility. | Clause 5.4 | m | y |
| 2 | NegSyn_0504_forbidden_parametrization_001 | The IUT rejects forbidden module parametrization types. | Clause 5.4 | m | n |
| 3 | NegSyn_0504_forbidden_parametrization_002 | The IUT rejects forbidden module parametrization types. | Clause 5.4 | m | y |

# 5.9. Formal parameters

*Table 8. Formal parameters*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_050401_top_level_001 | verify that error is generated for incompatible actual value of `in` parameter | Clause 5.4.1 | m | y |
| 2 | NegSem_050401_top_level_002 | verify that error is generated for incompatible actual value of `out` parameter | Clause 5.4.1 | m | y |
| 3 | NegSem_050401_top_level_003 | verify that error is generated if actual `inout` parameter doesn't adhere to strong typing rules | Clause 5.4.1 | m | n |
| 4 | Sem_050401_top_level_001 | verify that `in` parameters can be read within parametrized content | Clause 5.4.1 | m | y |
| 5 | Sem_050401_top_level_002 | verify that `out` parameters can be read within parametrized content | Clause 5.4.1 | m | n |
| 6 | Sem_050401_top_level_003 | verify that `inout` parameters can be read within parametrized content | Clause 5.4.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Sem_050401_top_level_004 | verify that `in` parameters can be set within parametrized content | Clause 5.4.1 | m | y |
| 8 | Sem_050401_top_level_005 | verify that `out` parameters can be set within parametrized content | Clause 5.4.1 | m | y |
| 9 | Sem_050401_top_level_006 | verify that `inout` parameters can be set within parametrized content | Clause 5.4.1 | m | y |
| 10 | Sem_050401_top_level_007 | verify that `in` parameters can be used as actual in parameters of parameterized objects | Clause 5.4.1 | m | y |
| 11 | Sem_050401_top_level_008 | verify that `in` parameters can be used as actual out parameters of parameterized objects | Clause 5.4.1 | m | y |
| 12 | Sem_050401_top_level_009 | verify that `in` parameters can be used as actual `inout` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 13 | Sem_050401_top_level_010 | verify that `out` parameters can be used as actual `in` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 14 | Sem_050401_top_level_011 | verify that `out` parameters can be used as actual `out` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 15 | Sem_050401_top_level_012 | verify that `out` parameters can be used as actual `inout` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 16 | Sem_050401_top_level_013 | verify that `inout` parameters can be used as actual `in` parameters of parameterized objects | Clause 5.4.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 17 | Sem_050401_top_level_014 | verify that `inout` parameters can be used as actual `out` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 18 | Sem_050401_top_level_015 | verify that `inout` parameters can be used as actual `inout` parameters of parameterized objects | Clause 5.4.1 | m | y |
| 19 | Sem_050401_top_level_016 | verify that compatibility rules are used for passing `in` parameters | Clause 5.4.1 | m | y |
| 20 | Sem_050401_top_level_017 | verify that compatibility rules are used for passing `out` parameters | Clause 5.4.1 | m | y |
| 21 | Sem_050401_top_level_018 | verify that strong typing is used for passing `inout` parameters | Clause 5.4.1 | m | y |
| 22 | Sem_050401_top_level_019 | verify that `@lazy` modifier can be used for value parameters | Clause 5.4.1 | m | y |
| 23 | Sem_050401_top_level_020 | verify that `@lazy` modifier can be used for template parameters | Clause 5.4.1 | m | y |
| 24 | Sem_050401_top_level_021 | verify that `@lazy` parameters containing component variable references are properly evaluated | Clause 5.4.1 | m | y |
| 25 | Sem_050401_top_level_022 | verify that `@fuzzy` modifier can be used for value parameters | Clause 5.4.1 | m | y |
| 26 | Sem_050401_top_level_023 | verify that `@fuzzy` modifier can be used for template parameters | Clause 5.4.1 | m | y |
| 27 | Sem_050401_top_level_024 | verify that `@fuzzy` parameters containing component variable references are properly evaluated | Clause 5.4.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 28 | Sem_050401_top_level_025 | verify that default values of `@lazy` parameters are properly evaluated | Clause 5.4.1 | m | y |
| 29 | Sem_050401_top_level_026 | verify that default values of `@fuzzy` parameters are properly evaluated | Clause 5.4.1 | m | n |
| 30 | Sem_050401_top_level_027 | verify that passing lazy parameter to formal parameter without modifier disables lazy evaluation | Clause 5.4.1 | m | y |
| 31 | Sem_050401_top_level_028 | verify that passing fuzzy parameter to formal parameter without modifier disables fuzzy evaluation | Clause 5.4.1 | m | y |
| 32 | Sem_050401_top_level_029 | verify that fuzzy parameter passed to lazy formal parameter enables lazy evaluation | Clause 5.4.1 | m | y |

# 5.10. Formal parameters of kind value

*Table 9. Formal parameters of kind value*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_05040101_parameters_of_kind_value_001 | verify that `in` value formal parameters of template cannot used dash as default value | Clause 5.4.1.1 | m | y |
| 2 | NegSem_05040101_parameters_of_kind_value_002 | verify that modified template cannot used dash as default value when original value parameter had no default value | Clause 5.4.1.1 | m | y |
| 3 | NegSem_05040101_parameters_of_kind_value_003 | verify that template definitions cannot contain `out` value formal parameters | Clause 5.4.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_05040101_parameters_of_kind_value_004 | verify that template definitions cannot contain `inout` value formal parameters | Clause 5.4.1.1 | m | y |
| 5 | NegSem_05040101_parameters_of_kind_value_005 | verify that `out` value formal parameters cannot have default values | Clause 5.4.1.1 | m | y |
| 6 | NegSem_05040101_parameters_of_kind_value_006 | verify that `inout` value formal parameters cannot have default values | Clause 5.4.1.1 | m | y |
| 7 | NegSem_05040101_parameters_of_kind_value_007 | verify that incompatible value in default value assignment of value formal parameters causes error | Clause 5.4.1.1 | m | y |
| 8 | NegSem_05040101_parameters_of_kind_value_008 | verify that default value of value formal parameters cannot reference component variables | Clause 5.4.1.1 | m | y |
| 9 | NegSem_05040101_parameters_of_kind_value_009 | verify that default value of value formal parameters cannot reference other parameters | Clause 5.4.1.1 | m | y |
| 10 | NegSem_05040101_parameters_of_kind_value_010 | verify that default value of value formal parameters cannot invoke functions with `runs on` clause | Clause 5.4.1.1 | m | y |
| 11 | NegSem_05040101_parameters_of_kind_value_011 | verify that error is generated if formal value parameter of function contains dash | Clause 5.4.1.1 | m | y |
| 12 | NegSem_05040101_parameters_of_kind_value_012 | verify that error is generated if formal value parameter of altstep contains dash | Clause 5.4.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | NegSem_05040101_parameters_of_kind_value_013 | verify that error is generated if formal value parameter of test case contains dash | Clause 5.4.1.1 | m | y |
| 14 | NegSem_05040101_parameters_of_kind_value_014 | verify that `out` formal value parameters cannot have lazy modifier | Clause 5.4.1.1 | m | y |
| 15 | NegSem_05040101_parameters_of_kind_value_015 | verify that `out` formal value parameters cannot have fuzzy modifier | Clause 5.4.1.1 | m | n |
| 16 | NegSem_05040101_parameters_of_kind_value_016 | verify that `inout` formal value parameters cannot have lazy modifier | Clause 5.4.1.1 | m | y |
| 17 | NegSem_05040101_parameters_of_kind_value_017 | verify that `inout` formal value parameters cannot have fuzzy modifier | Clause 5.4.1.1 | m | n |
| 18 | NegSyn_05040101_parameters_of_kind_value_001 | verify that `const` definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 19 | NegSyn_05040101_parameters_of_kind_value_002 | verify that `var` definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 20 | NegSyn_05040101_parameters_of_kind_value_003 | verify that template variable definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 21 | NegSyn_05040101_parameters_of_kind_value_004 | verify that timer definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 22 | NegSyn_05040101_parameters_of_kind_value_005 | verify that control definition cannot be parameterized | Clause 5.4.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | NegSyn_05040101_parameters_of_kind_value_006 | verify that record of definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 24 | NegSyn_05040101_parameters_of_kind_value_007 | verify that set of definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 25 | NegSyn_05040101_parameters_of_kind_value_008 | verify that enumerated definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 26 | NegSyn_05040101_parameters_of_kind_value_009 | verify that port definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 27 | NegSyn_05040101_parameters_of_kind_value_010 | verify that component definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 28 | NegSyn_05040101_parameters_of_kind_value_011 | verify that subtype definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 29 | NegSyn_05040101_parameters_of_kind_value_012 | verify that group definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 30 | NegSyn_05040101_parameters_of_kind_value_013 | verify that import definition cannot be parameterized | Clause 5.4.1.1 | m | y |
| 31 | Sem_05040101_parameters_of_kind_value_001 | The IUT correctly handles parametrization through the use of module parameters. | Clause 5.4.1.1 | m | y |
| 32 | Sem_05040101_parameters_of_kind_value_002 | The IUT correctly handles parametrization through the use of module parameters. | Clause 5.4.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 33 | Sem_05040101_parameters_of_kind_value_003 | The IUT correctly handles parametrization through the use of module parameters. | Clause 5.4.1.1 | m | y |
| 34 | Sem_05040101_parameters_of_kind_value_004 | The IUT correctly handles parametrization through the use of module parameters. | Clause 5.4.1.1 | m | y |
| 35 | Sem_05040101_parameters_of_kind_value_005 | verify that template definition can contain `in` value formal parameters | Clause 5.4.1.1 | m | y |
| 36 | Sem_05040101_parameters_of_kind_value_006 | verify that local template definition can contain `in` value formal parameters | Clause 5.4.1.1 | m | n |
| 37 | Sem_05040101_parameters_of_kind_value_007 | verify that function definition can contain `in`, `out` and `inout` value formal parameters | Clause 5.4.1.1 | m | y |
| 38 | Sem_05040101_parameters_of_kind_value_008 | verify that altstep definition can contain `in`, `out` and `inout` value formal parameters | Clause 5.4.1.1 | m | y |
| 39 | Sem_05040101_parameters_of_kind_value_009 | verify that test case definition can contain `in`, `out` and `inout` value formal parameters | Clause 5.4.1.1 | m | y |
| 40 | Sem_05040101_parameters_of_kind_value_010 | verify that value formal parameters can be used in expressions | Clause 5.4.1.1 | m | y |
| 41 | Sem_05040101_parameters_of_kind_value_011 | verify that `in` value formal parameters of template can have default values | Clause 5.4.1.1 | m | n |
| 42 | Sem_05040101_parameters_of_kind_value_012 | verify that `in` value formal parameters of local template can have default values | Clause 5.4.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 43 | Sem_05040101_parameters_of_kind_value_013 | verify that `in` value formal parameters of function can have default values | Clause 5.4.1.1 | m | y |
| 44 | Sem_05040101_parameters_of_kind_value_014 | verify that `in` value formal parameters of altstep can have default values | Clause 5.4.1.1 | m | y |
| 45 | Sem_05040101_parameters_of_kind_value_015 | verify that `in` value formal parameters of test case can have default values | Clause 5.4.1.1 | m | y |
| 46 | Sem_05040101_parameters_of_kind_value_016 | verify that `in` value formal parameters of modified template can used dash as default value | Clause 5.4.1.1 | m | y |
| 47 | Sem_05040101_parameters_of_kind_value_017 | verify that `null` is suitable default value of formal value parameters of component type | Clause 5.4.1.1 | m | y |
| 48 | Sem_05040101_parameters_of_kind_value_018 | verify that `self` is suitable default value of formal value parameters of component type | Clause 5.4.1.1 | m | n |
| 49 | Sem_05040101_parameters_of_kind_value_019 | verify that `mtc` is suitable default value of formal value parameters of component type | Clause 5.4.1.1 | m | y |
| 50 | Sem_05040101_parameters_of_kind_value_020 | verify that `system` is suitable default value of formal value parameters of component type | Clause 5.4.1.1 | m | y |
| 51 | Sem_05040101_parameters_of_kind_value_021 | verify that `null` can be used as default value of formal value parameters of default type | Clause 5.4.1.1 | m | y |
| 52 | Sem_05040101_parameters_of_kind_value_022 | verify that passing by value and by reference works correctly | Clause 5.4.1.1 | m | y |

# 5.11. Formal parameters of kind template

*Table 10. Formal parameters of kind template*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_05040102_parameters_of_kind_template_001 | verify that `in` template formal parameters of template cannot used dash as default value | Clause 5.4.1.2 | m | y |
| 2 | NegSem_05040102_parameters_of_kind_template_002 | verify that modified template cannot used dash as default value when original template parameter had no default value | Clause 5.4.1.2 | m | y |
| 3 | NegSem_05040102_parameters_of_kind_template_003 | verify that template definitions cannot contain `out` template formal parameters | Clause 5.4.1.2 | m | y |
| 4 | NegSem_05040102_parameters_of_kind_template_004 | verify that template definitions cannot contain `inout` template formal parameters | Clause 5.4.1.2 | m | y |
| 5 | NegSem_05040102_parameters_of_kind_template_005 | verify that `out` template formal parameters cannot have default values | Clause 5.4.1.2 | m | y |
| 6 | NegSem_05040102_parameters_of_kind_template_006 | verify that `inout` template formal parameters cannot have default values | Clause 5.4.1.2 | m | y |
| 7 | NegSem_05040102_parameters_of_kind_template_007 | verify that incompatible template instance in default template assignment of template formal parameters causes error | Clause 5.4.1.2 | m | y |
| 8 | NegSem_05040102_parameters_of_kind_template_008 | verify that default template instance of template formal parameters cannot reference component elements | Clause 5.4.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_05040102_parameters_of_kind_template_009 | verify that default template instance of template formal parameters cannot reference other parameters | Clause 5.4.1.2 | m | y |
| 10 | NegSem_05040102_parameters_of_kind_template_010 | verify that default template instance of template formal parameters cannot invoke functions with `runs on` clause | Clause 5.4.1.2 | m | y |
| 11 | NegSem_05040102_parameters_of_kind_template_011 | verify that error is generated if formal template parameter of function contains dash | Clause 5.4.1.2 | m | n |
| 12 | NegSem_05040102_parameters_of_kind_template_012 | verify that error is generated if formal template parameter of altstep contains dash | Clause 5.4.1.2 | m | n |
| 13 | NegSem_05040102_parameters_of_kind_template_013 | verify that error is generated if formal template parameter of test case contains dash | Clause 5.4.1.2 | m | n |
| 14 | NegSem_05040102_parameters_of_kind_template_014 | verify that out formal template parameters cannot have lazy modifier | Clause 5.4.1.2 | m | y |
| 15 | NegSem_05040102_parameters_of_kind_template_015 | verify that out formal template parameters cannot have fuzzy modifier | Clause 5.4.1.2 | m | n |
| 16 | NegSem_05040102_parameters_of_kind_template_016 | verify that `inout` formal template parameters cannot have lazy modifier | Clause 5.4.1.2 | m | y |
| 17 | NegSem_05040102_parameters_of_kind_template_017 | verify that `inout` formal template parameters cannot have fuzzy modifier | Clause 5.4.1.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | NegSem_05040102_parameters_of_kind_template_018 | Verify that template parameter of an activated altstep cannot be an out parameter | Clause 5.4.1.2 | m | n |
| 19 | NegSem_05040102_parameters_of_kind_template_019 | Verify that template parameter of an activated altstep cannot be an `inout` parameter | Clause 5.4.1.2 | m | n |
| 20 | NegSyn_05040102_parameters_of_kind_template_001 | verify that module parameter of template kind is not allowed | Clause 5.4.1.2 | m | n |
| 21 | Sem_05040102_parameters_of_kind_template_001 | The IUT correctly handles parametrization through the use of parameterized templates. | Clause 5.4.1.2 | m | y |
| 22 | Sem_05040102_parameters_of_kind_template_002 | The IUT correctly handles parametrization through the use of parameterized templates. | Clause 5.4.1.2 | m | y |
| 23 | Sem_05040102_parameters_of_kind_template_003 | verify that template definition can contain `in` template formal parameters | Clause 5.4.1.2 | m | y |
| 24 | Sem_05040102_parameters_of_kind_template_004 | verify that local template definition can contain `in` template formal parameters | Clause 5.4.1.2 | m | n |
| 25 | Sem_05040102_parameters_of_kind_template_005 | verify that function definition can contain `in`, `out` and `inout` template formal parameters | Clause 5.4.1.2 | m | y |
| 26 | Sem_05040102_parameters_of_kind_template_006 | verify that altstep definition can contain `in`, `out` and `inout` template formal parameters | Clause 5.4.1.2 | m | y |
| 27 | Sem_05040102_parameters_of_kind_template_007 | verify that test case definition can contain `in`, `out` and `inout` template formal parameters | Clause 5.4.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 28 | Sem_05040102_parameters_of_kind_template_008 | verify that template formal parameters can be used in the same way as templates or template variables | Clause 5.4.1.2 | m | y |
| 29 | Sem_05040102_parameters_of_kind_template_009 | verify that in template formal parameters of template can have default values | Clause 5.4.1.2 | m | y |
| 30 | Sem_05040102_parameters_of_kind_template_010 | verify that in template formal parameters of local template can have default values | Clause 5.4.1.2 | m | n |
| 31 | Sem_05040102_parameters_of_kind_template_011 | verify that in template formal parameters of function can have default values | Clause 5.4.1.2 | m | y |
| 32 | Sem_05040102_parameters_of_kind_template_012 | verify that in template formal parameters of altstep can have default values | Clause 5.4.1.2 | m | y |
| 33 | Sem_05040102_parameters_of_kind_template_013 | verify that in template formal parameters of test case can have default values | Clause 5.4.1.2 | m | y |
| 34 | Sem_05040102_parameters_of_kind_template_014 | verify that in template formal parameters of modified template can used dash as default value | Clause 5.4.1.2 | m | y |
| 35 | Sem_05040102_parameters_of_kind_template_015 | verify that template definition can contain in template formal parameters with omit restriction | Clause 5.4.1.2 | m | y |
| 36 | Sem_05040102_parameters_of_kind_template_016 | verify that local template definition can contain in template formal parameters with omit restriction | Clause 5.4.1.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 37 | Sem_05040102_parameters_of_kind_template_017 | verify that function definition can contain `in`, `out` and `inout` template formal parameters with omit restriction | Clause 5.4.1.2 | m | y |
| 38 | Sem_05040102_parameters_of_kind_template_018 | verify that altstep definition can contain `in`, `out` and `inout` template formal parameters with omit restriction | Clause 5.4.1.2 | m | y |
| 39 | Sem_05040102_parameters_of_kind_template_019 | verify that test case definition can contain `in`, `out` and `inout` template formal parameters with omit restriction | Clause 5.4.1.2 | m | y |
| 40 | Sem_05040102_parameters_of_kind_template_020 | verify that template definition can contain `in` template formal parameters with present restriction | Clause 5.4.1.2 | m | y |
| 41 | Sem_05040102_parameters_of_kind_template_021 | verify that local template definition can contain `in` template formal parameters with present restriction | Clause 5.4.1.2 | m | n |
| 42 | Sem_05040102_parameters_of_kind_template_022 | verify that function definition can contain `in`, `out` and `inout` template formal parameters with present restriction | Clause 5.4.1.2 | m | y |
| 43 | Sem_05040102_parameters_of_kind_template_023 | verify that altstep definition can contain `in`, `out` and `inout` template formal parameters with present restriction | Clause 5.4.1.2 | m | y |
| 44 | Sem_05040102_parameters_of_kind_template_024 | verify that test case definition can contain `in`, `out` and `inout` template formal parameters with present restriction | Clause 5.4.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 45 | Sem_05040102_parameters_of_kind_template_025 | verify that template definition can contain `in` template formal parameters with value restriction | Clause 5.4.1.2 | m | y |
| 46 | Sem_05040102_parameters_of_kind_template_026 | verify that local template definition can contain `in` template formal parameters with value restriction | Clause 5.4.1.2 | m | n |
| 47 | Sem_05040102_parameters_of_kind_template_027 | verify that function definition can contain in, out and `inout` template formal parameters with value restriction | Clause 5.4.1.2 | m | y |
| 48 | Sem_05040102_parameters_of_kind_template_028 | verify that altstep definition can contain in, out and `inout` template formal parameters with value restriction | Clause 5.4.1.2 | m | y |
| 49 | Sem_05040102_parameters_of_kind_template_029 | verify that test case definition can contain in, out and `inout` template formal parameters with value restriction | Clause 5.4.1.2 | m | y |
| 50 | Sem_05040102_parameters_of_kind_template_030 | verify that template definition can contain in template formal parameters with short omit restriction | Clause 5.4.1.2 | m | y |
| 51 | Sem_05040102_parameters_of_kind_template_031 | verify that local template definition can contain in template formal parameters with short omit restriction | Clause 5.4.1.2 | m | n |
| 52 | Sem_05040102_parameters_of_kind_template_032 | verify that function definition can contain in, out and `inout` template formal parameters with short omit restriction | Clause 5.4.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 53 | Sem_05040102_parameters_of_kind_template_033 | verify that altstep definition can contain in, out and `inout` template formal parameters with short omit restriction | Clause 5.4.1.2 | m | y |
| 54 | Sem_05040102_parameters_of_kind_template_034 | verify that test case definition can contain in, out and `inout` template formal parameters with short omit restriction | Clause 5.4.1.2 | m | y |
| 55 | Sem_05040102_parameters_of_kind_template_035 | verify that `null` is suitable default value of formal template parameters of component type | Clause 5.4.1.2 | m | y |
| 56 | Sem_05040102_parameters_of_kind_template_036 | verify that `self` is suitable default value of formal template parameters of component type | Clause 5.4.1.2 | m | n |
| 57 | Sem_05040102_parameters_of_kind_template_037 | verify that `mtc` is suitable default value of formal template parameters of component type | Clause 5.4.1.2 | m | y |
| 58 | Sem_05040102_parameters_of_kind_template_038 | verify that `system` is suitable default value of formal template parameters of component type | Clause 5.4.1.2 | m | y |

# 5.12. Formal parameters of kind timer

*Table 11. Formal parameters of kind timer*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_05040103_parameters_of_kind_timer_001 | Verify that functions with timer parameters cannot be used in `component.start` operation | Clause 5.4.1.3 | m | y |
| 2 | NegSem_05040103_parameters_of_kind_timer_002 | Verify that altsteps with timer parameters cannot be used in `component.start` operation | Clause 5.4.1.3 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_05040103_parameters_of_kind_timer_003 | Verify that test cases cannot have timer parameters | Clause 5.4.1.3 | m | y |
| 4 | NegSem_05040103_parameters_of_kind_timer_004 | Verify that templates cannot have timer parameters | Clause 5.4.1.3 | m | y |
| 5 | NegSyn_05040103_parameters_of_kind_timer_001 | Verify that in timer parameters are not allowed | Clause 5.4.1.3 | m | y |
| 6 | NegSyn_05040103_parameters_of_kind_timer_002 | Verify that out timer parameters are not allowed | Clause 5.4.1.3 | m | y |
| 7 | Sem_05040103_parameters_of_kind_timer_001 | The IUT correctly handles parametrization through the use of timer parameters. | Clause 5.4.1.3 | m | y |
| 8 | Sem_05040103_parameters_of_kind_timer_002 | Verify that `inout` prefix can be used for timer parameters | Clause 5.4.1.3 | m | y |
| 9 | Sem_05040103_parameters_of_kind_timer_003 | Verify that altstep can have timer parameters | Clause 5.4.1.3 | m | y |

# 5.13. Formal parameters of kind port

*Table 12. Formal parameters of kind port*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_05040104_parameters_of_kind_port_001 | Verify that functions with port parameters cannot be used in `component.start` operation | Clause 5.4.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_05040104_parameters_of_kind_port_002 | Verify that altsteps with port parameters cannot be used in `component.start` operation | Clause 5.4.1.4 | m | n |
| 3 | NegSem_05040104_parameters_of_kind_port_003 | Verify that in port parameters are not allowed | Clause 5.4.1.4 | m | y |
| 4 | NegSem_05040104_parameters_of_kind_port_004 | Verify that out port parameters are not allowed | Clause 5.4.1.4 | m | y |
| 5 | NegSem_05040104_parameters_of_kind_port_005 | Verify that test cases cannot have port parameters | Clause 5.4.1.4 | m | y |
| 6 | NegSem_05040104_parameters_of_kind_port_006 | Verify that templates cannot contain port parameters | Clause 5.4.1.4 | m | y |
| 7 | Sem_05040104_parameters_of_kind_port_001 | The IUT accepts port parametrization types for functions. | Clause 5.4.1.4 | m | y |
| 8 | Sem_05040104_parameters_of_kind_port_002 | Verify that `inout` prefix can be used for port parameters | Clause 5.4.1.4 | m | y |

# 5.14. Actual parameters

*Table 13. Actual parameters*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_050402_actual_parameters_001 | verify that template parameters cannot be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_050402_actual_parameters_002 | verify that template variables cannot be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 3 | NegSem_050402_actual_parameters_003 | verify that template `in` parameters cannot be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 4 | NegSem_050402_actual_parameters_004 | verify that template `out` parameters cannot be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 5 | NegSem_050402_actual_parameters_005 | verify that template `inout` parameters cannot be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 6 | NegSem_050402_actual_parameters_006 | verify that template parameters cannot be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 7 | NegSem_050402_actual_parameters_007 | verify that template variables cannot be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 8 | NegSem_050402_actual_parameters_008 | verify that template `in` parameters cannot be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 9 | NegSem_050402_actual_parameters_009 | verify that template `out` parameters cannot be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 10 | NegSem_050402_actual_parameters_010 | verify that template `inout` parameters cannot be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 11 | NegSem_050402_actual_parameters_011 | verify that template parameters cannot be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 12 | NegSem_050402_actual_parameters_012 | verify that template variables cannot be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 13 | NegSem_050402_actual_parameters_013 | verify that template `in` parameters cannot be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 14 | NegSem_050402_actual_parameters_014 | verify that template `out` parameters cannot be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 15 | NegSem_050402_actual_parameters_015 | verify that template `inout` parameters cannot be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 16 | NegSem_050402_actual_parameters_016 | verify that template parameters cannot be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 17 | NegSem_050402_actual_parameters_017 | verify that template variables cannot be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 18 | NegSem_050402_actual_parameters_018 | verify that template `in` parameters cannot be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 19 | NegSem_050402_actual_parameters_019 | verify that template `out` parameters cannot be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 20 | NegSem_050402_actual_parameters_020 | verify that template `inout` parameters cannot be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 21 | NegSem_050402_actual_parameters_021 | verify that literals cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 22 | NegSem_050402_actual_parameters_022 | verify that module parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 23 | NegSem_050402_actual_parameters_023 | verify that constants cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 24 | NegSem_050402_actual_parameters_024 | verify that function calls cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 25 | NegSem_050402_actual_parameters_025 | verify that expressions cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 26 | NegSem_050402_actual_parameters_026 | verify that template parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 27 | NegSem_050402_actual_parameters_027 | verify that template variables cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 28 | NegSem_050402_actual_parameters_028 | verify that template `in` parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 29 | NegSem_050402_actual_parameters_029 | verify that template `out` parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 30 | NegSem_050402_actual_parameters_030 | verify that template `inout` parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 31 | NegSem_050402_actual_parameters_031 | verify that template variable element reference cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 32 | NegSem_050402_actual_parameters_032 | verify that reference to elements of formal value parameters cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 33 | NegSem_050402_actual_parameters_033 | verify that literals cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 34 | NegSem_050402_actual_parameters_034 | verify that module parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 35 | NegSem_050402_actual_parameters_035 | verify that constants cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 36 | NegSem_050402_actual_parameters_036 | verify that function calls cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 37 | NegSem_050402_actual_parameters_037 | verify that expressions cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 38 | NegSem_050402_actual_parameters_038 | verify that template parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 39 | NegSem_050402_actual_parameters_039 | verify that template variables cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 40 | NegSem_050402_actual_parameters_040 | verify that template `in` parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 41 | NegSem_050402_actual_parameters_041 | verify that template out parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 42 | NegSem_050402_actual_parameters_042 | verify that template `inout` parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 43 | NegSem_050402_actual_parameters_043 | verify that template variable element reference cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 44 | NegSem_050402_actual_parameters_044 | verify that reference to elements of formal value parameters cannot be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 45 | NegSem_050402_actual_parameters_045 | verify that literals cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 46 | NegSem_050402_actual_parameters_046 | verify that module parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 47 | NegSem_050402_actual_parameters_047 | verify that constants cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 48 | NegSem_050402_actual_parameters_048 | verify that function calls cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 49 | NegSem_050402_actual_parameters_049 | verify that expressions cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 50 | NegSem_050402_actual_parameters_050 | verify that template parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 51 | NegSem_050402_actual_parameters_051 | verify that template variables cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 52 | NegSem_050402_actual_parameters_052 | verify that template `in` parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 53 | NegSem_050402_actual_parameters_053 | verify that template `out` parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 54 | NegSem_050402_actual_parameters_054 | verify that template `inout` parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 55 | NegSem_050402_actual_parameters_055 | verify that template variable element reference cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 56 | NegSem_050402_actual_parameters_056 | verify that reference to elements of formal value parameters cannot be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 57 | NegSem_050402_actual_parameters_057 | verify that literals cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 58 | NegSem_050402_actual_parameters_058 | verify that module parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 59 | NegSem_050402_actual_parameters_059 | verify that constants cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 60 | NegSem_050402_actual_parameters_060 | verify that function calls cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 61 | NegSem_050402_actual_parameters_061 | verify that expressions cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 62 | NegSem_050402_actual_parameters_062 | verify that template parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 63 | NegSem_050402_actual_parameters_063 | verify that literals cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 64 | NegSem_050402_actual_parameters_064 | verify that module parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 65 | NegSem_050402_actual_parameters_065 | verify that constants cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 66 | NegSem_050402_actual_parameters_066 | verify that function calls cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 67 | NegSem_050402_actual_parameters_067 | verify that expressions cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 68 | NegSem_050402_actual_parameters_068 | verify that template parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 69 | NegSem_050402_actual_parameters_069 | verify that literals cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 70 | NegSem_050402_actual_parameters_070 | verify that module parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 71 | NegSem_050402_actual_parameters_071 | verify that constants cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 72 | NegSem_050402_actual_parameters_072 | verify that function calls cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 73 | NegSem_050402_actual_parameters_073 | verify that expressions cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 74 | NegSem_050402_actual_parameters_074 | verify that template parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 75 | NegSem_050402_actual_parameters_075 | verify that literals cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 76 | NegSem_050402_actual_parameters_076 | verify that module parameters cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 77 | NegSem_050402_actual_parameters_077 | verify that constants cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 78 | NegSem_050402_actual_parameters_078 | verify that function calls cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 79 | NegSem_050402_actual_parameters_079 | verify that expressions cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 80 | NegSem_050402_actual_parameters_080 | verify that template parameters cannot be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 81 | NegSem_050402_actual_parameters_081 | verify that literals cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 82 | NegSem_050402_actual_parameters_082 | verify that module parameters cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 83 | NegSem_050402_actual_parameters_083 | verify that constants cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 84 | NegSem_050402_actual_parameters_084 | verify that function calls cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 85 | NegSem_050402_actual_parameters_085 | verify that expressions cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 86 | NegSem_050402_actual_parameters_086 | verify that template parameters cannot be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 87 | NegSem_050402_actual_parameters_087 | verify that literals cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 88 | NegSem_050402_actual_parameters_088 | verify that module parameters cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 89 | NegSem_050402_actual_parameters_089 | verify that constants cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 90 | NegSem_050402_actual_parameters_090 | verify that function calls cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 91 | NegSem_050402_actual_parameters_091 | verify that expressions cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 92 | NegSem_050402_actual_parameters_092 | verify that template parameters cannot be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 93 | NegSem_050402_actual_parameters_093 | verify that referencing errors are detected in actual parameters passed to `in` formal value parameters | Clause 5.4.2 | m | y |
| 94 | NegSem_050402_actual_parameters_094 | verify that referencing errors are detected in actual parameters passed to `in` formal template parameters | Clause 5.4.2 | m | y |
| 95 | NegSem_050402_actual_parameters_095 | verify that referencing errors are detected in actual parameters passed to `out` formal template parameters | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 96 | NegSem_050402_actual_parameters_096 | verify that referencing rules are correctly applied to actual parameters of `inout` formal template parameters | Clause 5.4.2 | m | y |
| 97 | NegSem_050402_actual_parameters_097 | verify that string item references cannot be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 98 | NegSem_050402_actual_parameters_098 | verify that ordinary values cannot be passed to timer parameters | Clause 5.4.2 | m | y |
| 99 | NegSem_050402_actual_parameters_099 | verify that values cannot be passed to port parameters | Clause 5.4.2 | m | y |
| 100 | NegSem_050402_actual_parameters_100 | verify that list notation containing actual parameters in wrong order is not accepted | Clause 5.4.2 | m | y |
| 101 | NegSem_050402_actual_parameters_101 | verify that list notation containing less actual parameters than required is not accepted | Clause 5.4.2 | m | y |
| 102 | NegSem_050402_actual_parameters_102 | verify that parameter without default value cannot be skipped | Clause 5.4.2 | m | y |
| 103 | NegSem_050402_actual_parameters_103 | verify that mixing list and assignment notation is not allowed in parameterized calls (value as actual parameter) | Clause 5.4.2 | m | y |
| 104 | NegSem_050402_actual_parameters_104 | verify that mixing list and assignment notation is not allowed in parameterized calls (skipped actual parameter) | Clause 5.4.2 | m | y |
| 105 | NegSem_050402_actual_parameters_105 | verify that parameters cannot be assigned more than once in assignment notation | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 106 | NegSem_050402_actual_parameters_106 | verify that assignment notation that doesn't contain all parameters is not accepted | Clause 5.4.2 | m | y |
| 107 | NegSem_050402_actual_parameters_107 | verify that incompatible values cannot be passed to in formal parameters | Clause 5.4.2 | m | y |
| 108 | NegSem_050402_actual_parameters_108 | verify that incompatible values cannot be passed from out formal parameters | Clause 5.4.2 | m | y |
| 109 | NegSem_050402_actual_parameters_109 | verify that incompatible values cannot be passed to `inout` formal parameters | Clause 5.4.2 | m | y |
| 110 | NegSem_050402_actual_parameters_110 | verify that values of compatible but distinct types cannot be passed to `inout` formal parameters | Clause 5.4.2 | m | n |
| 111 | NegSem_050402_actual_parameters_111 | verify that incompatible templates cannot be passed to template parameters with omit restriction | Clause 5.4.2 | m | y |
| 112 | NegSem_050402_actual_parameters_112 | verify that compatible templates can be passed to template parameters with value restriction | Clause 5.4.2 | m | y |
| 113 | NegSem_050402_actual_parameters_113 | verify that compatible templates can be passed to template parameters with present restriction | Clause 5.4.2 | m | y |
| 114 | NegSem_050402_actual_parameters_114 | verify that parametrized entities used as actual parameter cannot be passed without parameter list | Clause 5.4.2 | m | y |
| 115 | NegSem_050402_actual_parameters_115 | verify that error is generated when no actual parameter list is used for functions with no parameters | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 116 | NegSem_050402_actual_parameters_116 | verify that error is generated when no actual parameter list is used for test cases with no parameters | Clause 5.4.2 | m | y |
| 117 | NegSem_050402_actual_parameters_117 | verify that error is generated when no actual parameter list is used for altsteps with no parameters | Clause 5.4.2 | m | y |
| 118 | NegSem_050402_actual_parameters_118 | verify that error is generated when empty actual parameter list is used for templates with no parameters | Clause 5.4.2 | m | y |
| 119 | NegSem_050402_actual_parameters_119 | verify that uninitialized values cannot be passed to in formal parameters | Clause 5.4.2 | m | n |
| 120 | NegSem_050402_actual_parameters_120 | verify that uninitialized values cannot be passed to `inout` formal parameters | Clause 5.4.2 | m | n |
| 121 | NegSem_050402_actual_parameters_121 | verify that function calls passed to lazy formal parameters cannot contain `inout` parameters | Clause 5.4.2 | m | n |
| 122 | NegSem_050402_actual_parameters_122 | verify that function calls passed to fuzzy formal parameters cannot contain `inout` parameters | Clause 5.4.2 | m | n |
| 123 | NegSem_050402_actual_parameters_123 | verify that function calls passed to lazy formal parameters cannot contain out parameters | Clause 5.4.2 | m | n |
| 124 | NegSem_050402_actual_parameters_124 | verify that function calls passed to fuzzy formal parameters cannot contain out parameters | Clause 5.4.2 | m | n |
| 125 | NegSem_050402_actual_parameters_125 | verify that error is generated when lazy variable is passed to `inout` formal parameter | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 126 | NegSem_050402_actual_parameters_126 | verify that error is generated when fuzzy variable is passed to `inout` formal parameter | Clause 5.4.2 | m | n |
| 127 | NegSem_050402_actual_parameters_127 | verify that error is generated when lazy variable is passed to out formal parameter | Clause 5.4.2 | m | n |
| 128 | NegSem_050402_actual_parameters_128 | verify that error is generated when fuzzy variable is passed to out formal parameter | Clause 5.4.2 | m | n |
| 129 | NegSem_050402_actual_parameters_129 | verify that error is generated when passing record and its field to `inout` parameters | Clause 5.4.2 | m | n |
| 130 | NegSem_050402_actual_parameters_130 | verify that error is generated when passing set and its field to `inout` parameters | Clause 5.4.2 | m | n |
| 131 | NegSem_050402_actual_parameters_131 | verify that error is generated when passing union and its element to `inout` parameters | Clause 5.4.2 | m | n |
| 132 | NegSem_050402_actual_parameters_132 | verify that error is generated when passing record of and its element to `inout` parameters | Clause 5.4.2 | m | n |
| 133 | NegSem_050402_actual_parameters_133 | verify that error is generated when passing set of and its element to `inout` parameters | Clause 5.4.2 | m | n |
| 134 | NegSem_050402_actual_parameters_134 | verify that error is generated when passing array and its element to `inout` parameters | Clause 5.4.2 | m | n |
| 135 | NegSem_050402_actual_parameters_135 | verify that error is generated when passing anytype value and its element to `inout` parameters | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 136 | NegSem_050402_actual_parameters_136 | verify that error is generated when passing record and its sub-elements to `inout` parameters | Clause 5.4.2 | m | n |
| 137 | NegSem_050402_actual_parameters_137 | verify that error is generated when passing set and its sub-field to `inout` parameters | Clause 5.4.2 | m | n |
| 138 | NegSem_050402_actual_parameters_138 | verify that error is generated when passing union and its sub-element to `inout` parameters | Clause 5.4.2 | m | n |
| 139 | NegSem_050402_actual_parameters_139 | verify that error is generated when passing record of and its sub-element to `inout` parameters | Clause 5.4.2 | m | n |
| 140 | NegSem_050402_actual_parameters_140 | verify that error is generated when passing set of and its sub-element to `inout` parameters | Clause 5.4.2 | m | n |
| 141 | NegSem_050402_actual_parameters_141 | verify that error is generated when passing array and its sub-element to `inout` parameters | Clause 5.4.2 | m | n |
| 142 | NegSem_050402_actual_parameters_142 | verify that error is generated when passing anytype value and its sub-element to `inout` parameters | Clause 5.4.2 | m | n |
| 143 | NegSem_050402_actual_parameters_143 | verify that error is generated when passing distinct union alternatives to `inout` parameters | Clause 5.4.2 | m | n |
| 144 | NegSem_050402_actual_parameters_144 | verify that error is generated when passing distinct union alternatives to `inout` parameters | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 145 | NegSem_050402_actual_parameters_145 | verify that the fourth part of the Example 3 produces the expected error | Clause 5.4.2 | m | n |
| 146 | NegSem_050402_actual_parameters_146 | verify that literal cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 147 | NegSem_050402_actual_parameters_147 | verify that expression cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 148 | NegSem_050402_actual_parameters_148 | verify that function calls cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 149 | NegSem_050402_actual_parameters_149 | verify that module parameters cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 150 | NegSem_050402_actual_parameters_150 | verify that templates cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 151 | NegSem_050402_actual_parameters_151 | verify that constants cannot be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 152 | NegSem_050402_actual_parameters_152 | verify that literal cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 153 | NegSem_050402_actual_parameters_153 | verify that expression cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 154 | NegSem_050402_actual_parameters_154 | verify that function calls cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 155 | NegSem_050402_actual_parameters_155 | verify that module parameters cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 156 | NegSem_050402_actual_parameters_156 | verify that templates cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 157 | NegSem_050402_actual_parameters_157 | verify that constants cannot be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 158 | NegSem_050402_actual_parameters_158 | verify that function cannot have more actual than formal parameters | Clause 5.4.2 | m | y |
| 159 | NegSem_050402_actual_parameters_159 | verify that templates cannot have more actual than formal parameters | Clause 5.4.2 | m | y |
| 160 | NegSem_050402_actual_parameters_160 | verify that altstep cannot have more actual than formal parameters | Clause 5.4.2 | m | y |
| 161 | NegSem_050402_actual_parameters_161 | verify that function testcase cannot have more actual than formal parameters | Clause 5.4.2 | m | y |
| 162 | NegSem_050402_actual_parameters_162 | verify that restricted template variables cannot be passed to unrestricted `inout` template parameters | Clause 5.4.2 | m | n |
| 163 | NegSem_050402_actual_parameters_163 | verify that unrestricted template variables cannot be passed to restricted `inout` template parameters | Clause 5.4.2 | m | n |
| 164 | NegSem_050402_actual_parameters_164 | verify that restricted template variables cannot be passed to `inout` template parameters with a different restriction | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 165 | NegSem_050402_actual_parameters_165 | verify that value variables cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 166 | NegSem_050402_actual_parameters_166 | verify that value `in` parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 167 | NegSem_050402_actual_parameters_167 | verify that value `out` parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 168 | NegSem_050402_actual_parameters_168 | verify that value `inout` parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 169 | NegSem_050402_actual_parameters_169 | verify that value variable element reference cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 170 | NegSem_050402_actual_parameters_170 | verify that reference to elements of formal value parameters cannot be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 171 | NegSem_050402_actual_parameters_171 | verify that value variables cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 172 | NegSem_050402_actual_parameters_172 | verify that value `in` parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 173 | NegSem_050402_actual_parameters_173 | verify that value `out` parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 174 | NegSem_050402_actual_parameters_174 | verify that value `inout` parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 175 | NegSem_050402_actual_parameters_175 | verify that value variable element reference cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 176 | NegSem_050402_actual_parameters_176 | verify that reference to elements of formal value parameters cannot be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 177 | NegSem_050402_actual_parameters_177 | verify that value variables cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 178 | NegSem_050402_actual_parameters_178 | verify that value `in` parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 179 | NegSem_050402_actual_parameters_179 | verify that value `in` parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 180 | NegSem_050402_actual_parameters_180 | verify that value `in` parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 181 | NegSem_050402_actual_parameters_181 | verify that value `in` parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 182 | NegSem_050402_actual_parameters_182 | verify that value `in` parameters cannot be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 183 | Sem_050402_actual_parameters_001 | The IUT accepts allowed assignments of actual parameters. | Clause 5.4.2 | m | y |
| 184 | Sem_050402_actual_parameters_002 | The IUT accepts nested assignment of actual parameters. | Clause 5.4.2 | m | y |
| 185 | Sem_050402_actual_parameters_003 | verify that literals can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 186 | Sem_050402_actual_parameters_004 | verify that module parameters can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 187 | Sem_050402_actual_parameters_005 | verify that constants can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 188 | Sem_050402_actual_parameters_006 | verify that variables can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 189 | Sem_050402_actual_parameters_007 | verify that function calls can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 190 | Sem_050402_actual_parameters_008 | verify that in value parameters can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 191 | Sem_050402_actual_parameters_009 | verify that out value parameters can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 192 | Sem_050402_actual_parameters_010 | verify that `inout` value parameters can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 193 | Sem_050402_actual_parameters_011 | verify that expressions can be used as `in` formal value parameters of functions | Clause 5.4.2 | m | y |
| 194 | Sem_050402_actual_parameters_012 | verify that literals can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 195 | Sem_050402_actual_parameters_013 | verify that module parameters can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 196 | Sem_050402_actual_parameters_014 | verify that constants can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 197 | Sem_050402_actual_parameters_015 | verify that variables can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 198 | Sem_050402_actual_parameters_016 | verify that function calls can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 199 | Sem_050402_actual_parameters_017 | verify that `in` value parameters can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 200 | Sem_050402_actual_parameters_018 | verify that out value parameters can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 201 | Sem_050402_actual_parameters_019 | verify that `inout` value parameters can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |
| 202 | Sem_050402_actual_parameters_020 | verify that expressions can be used as `in` formal value parameters of templates | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 203 | Sem_050402_actual_parameters_021 | verify that literals can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 204 | Sem_050402_actual_parameters_022 | verify that module parameters can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 205 | Sem_050402_actual_parameters_023 | verify that constants can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 206 | Sem_050402_actual_parameters_024 | verify that variables can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 207 | Sem_050402_actual_parameters_025 | verify that function calls can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 208 | Sem_050402_actual_parameters_026 | verify that `in` value parameters can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 209 | Sem_050402_actual_parameters_027 | verify that out value parameters can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 210 | Sem_050402_actual_parameters_028 | verify that `inout` value parameters can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 211 | Sem_050402_actual_parameters_029 | verify that expressions can be used as `in` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 212 | Sem_050402_actual_parameters_030 | verify that literals can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 213 | Sem_050402_actual_parameters_031 | verify that module parameters can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 214 | Sem_050402_actual_parameters_032 | verify that constants can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 215 | Sem_050402_actual_parameters_033 | verify that variables can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 216 | Sem_050402_actual_parameters_034 | verify that function calls can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 217 | Sem_050402_actual_parameters_035 | verify that `in` value parameters can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 218 | Sem_050402_actual_parameters_036 | verify that `out` value parameters can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 219 | Sem_050402_actual_parameters_037 | verify that `inout` value parameters can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 220 | Sem_050402_actual_parameters_038 | verify that expressions can be used as `in` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 221 | Sem_050402_actual_parameters_039 | verify that variables can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 222 | Sem_050402_actual_parameters_040 | verify that `in` value parameters can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 223 | Sem_050402_actual_parameters_041 | verify that `out` value parameters can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 224 | Sem_050402_actual_parameters_042 | verify that `inout` value parameters can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 225 | Sem_050402_actual_parameters_043 | verify that variable element reference can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 226 | Sem_050402_actual_parameters_044 | verify that reference to elements of formal value parameters can be used as `inout` formal value parameters of functions | Clause 5.4.2 | m | y |
| 227 | Sem_050402_actual_parameters_045 | verify that variables can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 228 | Sem_050402_actual_parameters_046 | verify that `in` value parameters can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 229 | Sem_050402_actual_parameters_047 | verify that `out` value parameters can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 230 | Sem_050402_actual_parameters_048 | verify that `inout` value parameters can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 231 | Sem_050402_actual_parameters_049 | verify that variable element reference can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |
| 232 | Sem_050402_actual_parameters_050 | verify that reference to elements of formal value parameters can be used as `inout` formal value parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 233 | Sem_050402_actual_parameters_051 | verify that variables can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 234 | Sem_050402_actual_parameters_052 | verify that `in` value parameters can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 235 | Sem_050402_actual_parameters_053 | verify that `out` value parameters can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 236 | Sem_050402_actual_parameters_054 | verify that `inout` value parameters can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 237 | Sem_050402_actual_parameters_055 | verify that variable element reference can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 238 | Sem_050402_actual_parameters_056 | verify that reference to elements of formal value parameters can be used as `inout` formal value parameters of test cases | Clause 5.4.2 | m | y |
| 239 | Sem_050402_actual_parameters_057 | verify that literals can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 240 | Sem_050402_actual_parameters_058 | verify that module parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 241 | Sem_050402_actual_parameters_059 | verify that constants can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 242 | Sem_050402_actual_parameters_060 | verify that variables can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 243 | Sem_050402_actual_parameters_061 | verify that function calls can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 244 | Sem_050402_actual_parameters_062 | verify that `in` value parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 245 | Sem_050402_actual_parameters_063 | verify that `out` value parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 246 | Sem_050402_actual_parameters_064 | verify that `inout` value parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 247 | Sem_050402_actual_parameters_065 | verify that expressions can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 248 | Sem_050402_actual_parameters_066 | verify that template parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 249 | Sem_050402_actual_parameters_067 | verify that template variables can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 250 | Sem_050402_actual_parameters_068 | verify that template `in` parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 251 | Sem_050402_actual_parameters_069 | verify that template `out` parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |
| 252 | Sem_050402_actual_parameters_070 | verify that template `inout` parameters can be used as in formal template parameters of functions | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 253 | Sem_050402_actual_parameters_071 | verify that literals can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 254 | Sem_050402_actual_parameters_072 | verify that module parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 255 | Sem_050402_actual_parameters_073 | verify that constants can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 256 | Sem_050402_actual_parameters_074 | verify that variables can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 257 | Sem_050402_actual_parameters_075 | verify that function calls can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 258 | Sem_050402_actual_parameters_076 | verify that `in` value parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 259 | Sem_050402_actual_parameters_077 | verify that `out` value parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 260 | Sem_050402_actual_parameters_078 | verify that `inout` value parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 261 | Sem_050402_actual_parameters_079 | verify that expressions can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 262 | Sem_050402_actual_parameters_080 | verify that template parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 263 | Sem_050402_actual_parameters_081 | verify that template variables can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 264 | Sem_050402_actual_parameters_082 | verify that template `in` parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 265 | Sem_050402_actual_parameters_083 | verify that template `out` parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 266 | Sem_050402_actual_parameters_084 | verify that template `inout` parameters can be used as in formal template parameters of templates | Clause 5.4.2 | m | y |
| 267 | Sem_050402_actual_parameters_085 | verify that literals can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 268 | Sem_050402_actual_parameters_086 | verify that module parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 269 | Sem_050402_actual_parameters_087 | verify that constants can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 270 | Sem_050402_actual_parameters_088 | verify that variables can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 271 | Sem_050402_actual_parameters_089 | verify that function calls can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 272 | Sem_050402_actual_parameters_090 | verify that `in` value parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 273 | Sem_050402_actual_parameters_091 | verify that `out` value parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 274 | Sem_050402_actual_parameters_092 | verify that `inout` value parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 275 | Sem_050402_actual_parameters_093 | verify that expressions can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 276 | Sem_050402_actual_parameters_094 | verify that template parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 277 | Sem_050402_actual_parameters_095 | verify that template variables can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 278 | Sem_050402_actual_parameters_096 | verify that template `in` parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 279 | Sem_050402_actual_parameters_097 | verify that template `out` parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 280 | Sem_050402_actual_parameters_098 | verify that template `inout` parameters can be used as in formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 281 | Sem_050402_actual_parameters_099 | verify that literals can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 282 | Sem_050402_actual_parameters_100 | verify that module parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 283 | Sem_050402_actual_parameters_101 | verify that constants can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 284 | Sem_050402_actual_parameters_102 | verify that variables can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 285 | Sem_050402_actual_parameters_103 | verify that function calls can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 286 | Sem_050402_actual_parameters_104 | verify that `in` value parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 287 | Sem_050402_actual_parameters_105 | verify that `out` value parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 288 | Sem_050402_actual_parameters_106 | verify that `inout` value parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 289 | Sem_050402_actual_parameters_107 | verify that expressions can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 290 | Sem_050402_actual_parameters_108 | verify that template parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 291 | Sem_050402_actual_parameters_109 | verify that template variables can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 292 | Sem_050402_actual_parameters_110 | verify that template `in` parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 293 | Sem_050402_actual_parameters_111 | verify that template `out` parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 294 | Sem_050402_actual_parameters_112 | verify that template `inout` parameters can be used as in formal template parameters of test cases | Clause 5.4.2 | m | y |
| 295 | Sem_050402_actual_parameters_113 | verify that template variables can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 296 | Sem_050402_actual_parameters_114 | verify that template `in` parameters can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 297 | Sem_050402_actual_parameters_115 | verify that template `out` parameters can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 298 | Sem_050402_actual_parameters_116 | verify that template `inout` parameters can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 299 | Sem_050402_actual_parameters_117 | verify that template variable element reference can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 300 | Sem_050402_actual_parameters_118 | verify that reference to elements of formal value parameters can be used as out formal template parameters of functions | Clause 5.4.2 | m | y |
| 301 | Sem_050402_actual_parameters_119 | verify that template variables can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 302 | Sem_050402_actual_parameters_120 | verify that template `in` parameters can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 303 | Sem_050402_actual_parameters_121 | verify that template `out` parameters can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 304 | Sem_050402_actual_parameters_122 | verify that template `inout` parameters can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 305 | Sem_050402_actual_parameters_123 | verify that template variable element reference can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 306 | Sem_050402_actual_parameters_124 | verify that reference to elements of formal value parameters can be used as out formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 307 | Sem_050402_actual_parameters_125 | verify that template variables can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 308 | Sem_050402_actual_parameters_126 | verify that template `in` parameters can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 309 | Sem_050402_actual_parameters_127 | verify that template `out` parameters can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 310 | Sem_050402_actual_parameters_128 | verify that template `inout` parameters can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 311 | Sem_050402_actual_parameters_129 | verify that template variable element reference can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 312 | Sem_050402_actual_parameters_130 | verify that reference to elements of formal value parameters can be used as out formal template parameters of test cases | Clause 5.4.2 | m | y |
| 313 | Sem_050402_actual_parameters_131 | verify that template variables can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 314 | Sem_050402_actual_parameters_132 | verify that template `in` parameters can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 315 | Sem_050402_actual_parameters_133 | verify that template `out` parameters can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 316 | Sem_050402_actual_parameters_134 | verify that template `inout` parameters can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 317 | Sem_050402_actual_parameters_135 | verify that template variable element reference can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 318 | Sem_050402_actual_parameters_136 | verify that reference to elements of formal value parameters can be used as `inout` formal template parameters of functions | Clause 5.4.2 | m | y |
| 319 | Sem_050402_actual_parameters_137 | verify that template variables can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 320 | Sem_050402_actual_parameters_138 | verify that template `in` parameters can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 321 | Sem_050402_actual_parameters_139 | verify that template `out` parameters can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 322 | Sem_050402_actual_parameters_140 | verify that template `inout` parameters can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 323 | Sem_050402_actual_parameters_141 | verify that template variable element reference can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 324 | Sem_050402_actual_parameters_142 | verify that reference to elements of formal value parameters can be used as `inout` formal template parameters of altsteps | Clause 5.4.2 | m | y |
| 325 | Sem_050402_actual_parameters_143 | verify that template variables can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 326 | Sem_050402_actual_parameters_144 | verify that template `in` parameters can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 327 | Sem_050402_actual_parameters_145 | verify that template `out` parameters can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 328 | Sem_050402_actual_parameters_146 | verify that template `inout` parameters can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 329 | Sem_050402_actual_parameters_147 | verify that template variable element reference can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 330 | Sem_050402_actual_parameters_148 | verify that reference to elements of formal value parameters can be used as `inout` formal template parameters of test cases | Clause 5.4.2 | m | y |
| 331 | Sem_050402_actual_parameters_149 | verify that referencing rules are correctly applied to actual parameters of `in` formal value parameters | Clause 5.4.2 | m | y |
| 332 | Sem_050402_actual_parameters_150 | verify that referencing rules are correctly applied to actual parameters of in formal template parameters | Clause 5.4.2 | m | n |
| 333 | Sem_050402_actual_parameters_151 | verify that referencing rules are correctly applied to actual parameters of `out` formal value parameters | Clause 5.4.2 | m | y |
| 334 | Sem_050402_actual_parameters_152 | verify that referencing rules are correctly applied to actual parameters of out formal template parameters | Clause 5.4.2 | m | y |
| 335 | Sem_050402_actual_parameters_153 | verify that referencing rules are correctly applied to actual parameters of `inout` formal value parameters | Clause 5.4.2 | m | y |
| 336 | Sem_050402_actual_parameters_154 | verify that referencing rules are correctly applied to actual parameters of `inout` formal template parameters | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 337 | Sem_050402_actual_parameters_155 | verify that `out` formal parameters are passed to actual parameter in correct (list notation) | Clause 5.4.2 | m | y |
| 338 | Sem_050402_actual_parameters_156 | verify that `out` formal parameters are passed to actual parameter in correct (assignment notation) | Clause 5.4.2 | m | n |
| 339 | Sem_050402_actual_parameters_157 | verify that component timers can be passed to timer parameters | Clause 5.4.2 | m | y |
| 340 | Sem_050402_actual_parameters_158 | verify that component timers can be passed to timer parameters | Clause 5.4.2 | m | y |
| 341 | Sem_050402_actual_parameters_159 | verify that timer parameters can be passed to timer parameters | Clause 5.4.2 | m | y |
| 342 | Sem_050402_actual_parameters_160 | verify that component ports can be passed to port parameters | Clause 5.4.2 | m | y |
| 343 | Sem_050402_actual_parameters_161 | verify that port parameters can be passed to port parameters | Clause 5.4.2 | m | y |
| 344 | Sem_050402_actual_parameters_162 | verify that actual parameters override default values | Clause 5.4.2 | m | y |
| 345 | Sem_050402_actual_parameters_163 | verify that default values are used if actual parameters are missing | Clause 5.4.2 | m | y |
| 346 | Sem_050402_actual_parameters_164 | verify that actual parameters override default templates | Clause 5.4.2 | m | y |
| 347 | Sem_050402_actual_parameters_165 | verify that default templates are used if actual parameters are missing | Clause 5.4.2 | m | y |
| 348 | Sem_050402_actual_parameters_166 | verify that actual parameters are evaluated in order of their appearance (list notation) | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 349 | Sem_050402_actual_parameters_167 | verify that actual parameters are evaluated in order of their appearance (assignment notation) | Clause 5.4.2 | m | n |
| 350 | Sem_050402_actual_parameters_168 | verify that rules for referencing are applied to actual paremeters before passing to out formal parameters | Clause 5.4.2 | m | y |
| 351 | Sem_050402_actual_parameters_169 | verify that rules for referencing are applied to actual paremeters before passing to `inout` formal parameters | Clause 5.4.2 | m | y |
| 352 | Sem_050402_actual_parameters_170 | verify that default parameters are evaluated in order of the formal parameter list (list notation) | Clause 5.4.2 | m | n |
| 353 | Sem_050402_actual_parameters_171 | verify that default parameters are evaluated in order of the formal parameter list (assignment notation) | Clause 5.4.2 | m | n |
| 354 | Sem_050402_actual_parameters_172 | verify that it is possible to use parametrized template with no parentheses if all parameters have default values | Clause 5.4.2 | m | y |
| 355 | Sem_050402_actual_parameters_173 | verify that it is possible to use parametrized template with empty parentheses | Clause 5.4.2 | m | y |
| 356 | Sem_050402_actual_parameters_174 | verify that actual parameter values override default values | Clause 5.4.2 | m | y |
| 357 | Sem_050402_actual_parameters_175 | verify that actual parameters in the beginning of list notation can be skipped | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 358 | Sem_050402_actual_parameters_176 | verify that multiple actual parameters of list notation can be skipped | Clause 5.4.2 | m | y |
| 359 | Sem_050402_actual_parameters_177 | verify that actual parameters at the end of list notation can be explicitly skipped | Clause 5.4.2 | m | y |
| 360 | Sem_050402_actual_parameters_178 | verify that missing actual parameters at the end of list notation are considered to be skipped (single parameter) | Clause 5.4.2 | m | y |
| 361 | Sem_050402_actual_parameters_179 | verify that missing actual parameters at the end of list notation are considered to be skipped (multiple parameter) | Clause 5.4.2 | m | y |
| 362 | Sem_050402_actual_parameters_180 | verify that assignment notation containing all parameters in declaration order is accepted | Clause 5.4.2 | m | y |
| 363 | Sem_050402_actual_parameters_181 | verify that assignment notation containing all parameters in random order is accepted | Clause 5.4.2 | m | n |
| 364 | Sem_050402_actual_parameters_182 | verify that assignment notation can omit parameters with default value | Clause 5.4.2 | m | y |
| 365 | Sem_050402_actual_parameters_183 | verify that compatible values can be passed to in formal parameters | Clause 5.4.2 | m | y |
| 366 | Sem_050402_actual_parameters_184 | verify that compatible values can be passed from out formal parameters | Clause 5.4.2 | m | y |
| 367 | Sem_050402_actual_parameters_185 | verify that compatible templates can be passed to template parameters with omit restriction | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 368 | Sem_050402_actual_parameters_186 | verify that compatible templates can be passed to template parameters with value restriction | Clause 5.4.2 | m | y |
| 369 | Sem_050402_actual_parameters_187 | verify that compatible templates can be passed to template parameters with present restriction | Clause 5.4.2 | m | y |
| 370 | Sem_050402_actual_parameters_188 | verify that it is possible to use nested actual parameter lists | Clause 5.4.2 | m | y |
| 371 | Sem_050402_actual_parameters_189 | verify that empty actual parameter list can be used for functions with no parameters | Clause 5.4.2 | m | y |
| 372 | Sem_050402_actual_parameters_190 | verify that empty actual parameter list can be used for altsteps with no parameters | Clause 5.4.2 | m | y |
| 373 | Sem_050402_actual_parameters_191 | verify that partially initialized values can be passed to in formal parameters | Clause 5.4.2 | m | y |
| 374 | Sem_050402_actual_parameters_192 | verify that partially initialized values can be passed to `inout` formal parameters | Clause 5.4.2 | m | y |
| 375 | Sem_050402_actual_parameters_193 | verify that Example 1 can be executed | Clause 5.4.2 | m | n |
| 376 | Sem_050402_actual_parameters_194 | verify that Example 2 can be executed | Clause 5.4.2 | m | y |
| 377 | Sem_050402_actual_parameters_195 | verify that the first part of the Example 3 can be executed | Clause 5.4.2 | m | y |
| 378 | Sem_050402_actual_parameters_196 | verify that the third part of the Example 3 can be executed | Clause 5.4.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 379 | Sem_050402_actual_parameters_198 | verify that the the Example 4 can be executed | Clause 5.4.2 | m | y |
| 380 | Sem_050402_actual_parameters_199 | verify that the Example 5 can be executed | Clause 5.4.2 | m | y |
| 381 | Sem_050402_actual_parameters_200 | verify that the Example 6 can be executed | Clause 5.4.2 | m | y |
| 382 | Sem_050402_actual_parameters_201 | verify that the Example 7 can be executed | Clause 5.4.2 | m | y |
| 383 | Sem_050402_actual_parameters_202 | verify that the Example 8 can be executed | Clause 5.4.2 | m | n |
| 384 | Sem_050402_actual_parameters_203 | verify that variables can be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 385 | Sem_050402_actual_parameters_204 | verify that variables can be used as actual out value parameters of functions | Clause 5.4.2 | m | n |
| 386 | Sem_050402_actual_parameters_205 | verify that in value parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 387 | Sem_050402_actual_parameters_206 | verify that `out` value parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 388 | Sem_050402_actual_parameters_207 | verify that `inout` value parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | y |
| 389 | Sem_050402_actual_parameters_208 | verify that `in` template parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 390 | Sem_050402_actual_parameters_209 | verify that `out` template parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | n |
| 391 | Sem_050402_actual_parameters_210 | verify that `inout` template parameters can be used as actual out value parameters of functions | Clause 5.4.2 | m | n |
| 392 | Sem_050402_actual_parameters_211 | verify that dash can be used as an actual out value parameter of functions | Clause 5.4.2 | m | n |
| 393 | Sem_050402_actual_parameters_212 | verify that variables can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 394 | Sem_050402_actual_parameters_213 | verify that variables can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | n |
| 395 | Sem_050402_actual_parameters_214 | verify that in value parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 396 | Sem_050402_actual_parameters_215 | verify that `out` value parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 397 | Sem_050402_actual_parameters_216 | verify that `inout` value parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | y |
| 398 | Sem_050402_actual_parameters_217 | verify that `in` template parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | n |
| 399 | Sem_050402_actual_parameters_218 | verify that `out` template parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 400 | Sem_050402_actual_parameters_219 | verify that `inout` template parameters can be used as actual out value parameters of altsteps | Clause 5.4.2 | m | n |
| 401 | Sem_050402_actual_parameters_220 | verify that dash can be used as an actual out value parameter of altsteps | Clause 5.4.2 | m | n |
| 402 | Sem_050402_actual_parameters_221 | verify that dash can be used as an actual out template parameter of functions | Clause 5.4.2 | m | n |
| 403 | Sem_050402_actual_parameters_222 | verify that dash can be used as an actual out template parameter of altsteps | Clause 5.4.2 | m | n |
| 404 | Sem_050402_actual_parameters_223 | verify that actual out value parameters of functions can be skipped if they are the last ones | Clause 5.4.2 | m | n |
| 405 | Sem_050402_actual_parameters_224 | verify that actual `out` value parameters of altsteps can be skipped if they are the last ones | Clause 5.4.2 | m | n |
| 406 | Sem_050402_actual_parameters_225 | verify that actual `out` template parameters of functions can be skipped if they are the last ones | Clause 5.4.2 | m | n |
| 407 | Sem_050402_actual_parameters_226 | verify that actual `out` template parameters of altsteps can be skipped if they are the last ones | Clause 5.4.2 | m | n |

## 5.15. Cyclic definitions

*Table 14. Cyclic definitions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0505_cyclic_definitions_001 | Verify that an error is detected when two constants reference each other | Clause 5.5 | m | y |
| 2 | NegSem_0505_cyclic_definitions_002 | Verify that an error is detected when a forbidded cyclic reference occurs in cyclic import | Clause 5.5 | m | y |
| 3 | Sem_0505_cyclic_definitions_001 | The IUT correctly handles recursive functions | Clause 5.5 | m | y |
| 4 | Sem_0505_cyclic_definitions_002 | The IUT correctly handles cyclic imports | Clause 5.5 | m | y |
| 5 | Sem_0505_cyclic_definitions_003 | Verify that cyclic import containing cyclic function calls is allowed | Clause 5.5 | m | y |
| 6 | Sem_0505_cyclic_definitions_004 | Verify that cyclic altsteps are allowed | Clause 5.5 | m | y |

# 5.16. Simple basic types and values

*Table 15. Simple basic types and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_060100_SimpleBasicTypes_001 | Assign float to integer values | Clause 6.1.0 | m | y |
| 2 | NegSyn_060100_SimpleBasicTypes_002 | Assign boolean to integer values | Clause 6.1.0 | m | y |
| 3 | NegSyn_060100_SimpleBasicTypes_003 | Assign integer to float values | Clause 6.1.0 | m | y |
| 4 | NegSyn_060100_SimpleBasicTypes_004 | Assign boolean to float values | Clause 6.1.0 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSyn_060100_SimpleBasicTypes_005 | Assign verdicttype to float values | Clause 6.1.0 | m | y |
| 6 | NegSyn_060100_SimpleBasicTypes_006 | Assign integer to verdicttype values | Clause 6.1.0 | m | y |
| 7 | Sem_060100_SimpleBasicTypes_001 | Assign and read integer values | Clause 6.1.0 | m | y |
| 8 | Sem_060100_SimpleBasicTypes_002 | Assign and read large integer values | Clause 6.1.0 | m | y |
| 9 | Sem_060100_SimpleBasicTypes_003 | Assign and read float values | Clause 6.1.0 | m | y |
| 10 | Sem_060100_SimpleBasicTypes_004 | Assign and read large float values | Clause 6.1.0 | m | y |
| 11 | Sem_060100_SimpleBasicTypes_005 | Assign and read verdicts | Clause 6.1.0 | m | y |
| 12 | Syn_060100_SimpleBasicTypes_001 | Assign different integer values | Clause 6.1.0 | m | y |
| 13 | Syn_060100_SimpleBasicTypes_002 | Assign large integer values | Clause 6.1.0 | m | y |
| 14 | Syn_060100_SimpleBasicTypes_003 | Assign different float values | Clause 6.1.0 | m | y |
| 15 | Syn_060100_SimpleBasicTypes_004 | Assign small and large float values | Clause 6.1.0 | m | y |
| 16 | Syn_060100_SimpleBasicTypes_005 | Accept float mantissa for float values | Clause 6.1.0 | m | y |
| 17 | Syn_060100_SimpleBasicTypes_006 | Accept all verdict values | Clause 6.1.0 | m | y |

# 5.17. Basic string types and values

*Table 16. Basic string types and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_060101_TopLevel_001 | Assign invalid bitstring value | Clause 6.1.1 | m | y |
| 2 | NegSyn_060101_TopLevel_002 | Assign string to bitstring values | Clause 6.1.1 | m | y |
| 3 | NegSyn_060101_TopLevel_003 | Assign octetstring to bitstring values | Clause 6.1.1 | m | y |
| 4 | NegSyn_060101_TopLevel_004 | Assign invalid hexstring value | Clause 6.1.1 | m | y |
| 5 | NegSyn_060101_TopLevel_005 | Assign string to hexstring values | Clause 6.1.1 | m | y |
| 6 | NegSyn_060101_TopLevel_006 | Assign octetstring to hexstring values | Clause 6.1.1 | m | y |
| 7 | NegSyn_060101_TopLevel_007 | Assign invalid octetstring value | Clause 6.1.1 | m | y |
| 8 | NegSyn_060101_TopLevel_008 | Assign string to octetstring values | Clause 6.1.1 | m | y |
| 9 | NegSyn_060101_TopLevel_009 | Assign hexstring to octetstring values | Clause 6.1.1 | m | y |
| 10 | NegSyn_060101_TopLevel_010 | Assign invalid hexstring value | Clause 6.1.1 | m | y |
| 11 | Sem_060101_TopLevel_001 | Assign and read bitstring | Clause 6.1.1 | m | y |
| 12 | Sem_060101_TopLevel_002 | Assign and read hexstring | Clause 6.1.1 | m | y |
| 13 | Sem_060101_TopLevel_003 | Assign and read octetstring | Clause 6.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_060101_TopLevel_004 | Assign and read charstring | Clause 6.1.1 | m | y |
| 15 | Sem_060101_TopLevel_005 | Assign and read universal charstring | Clause 6.1.1 | m | y |
| 16 | Sem_060101_TopLevel_006 | Assign and read universal charstring | Clause 6.1.1 | m | y |
| 17 | Sem_060101_TopLevel_007 | Assign and read universal charstring using USI like notation | Clause 6.1.1 | m | y |
| 18 | Sem_060101_TopLevel_008 | Assign and read bitstring with newline character | Clause 6.1.1 | m | n |
| 19 | Sem_060101_TopLevel_009 | Whitespaces, control characters and backslash will be ignored for the bitstring length calculation | Clause 6.1.1 | m | n |
| 20 | Sem_060101_TopLevel_010 | Assign and read hexstring with newline character | Clause 6.1.1 | m | n |
| 21 | Sem_060101_TopLevel_011 | Whitespaces, control characters and backslash will be ignored for the hexstring length calculation | Clause 6.1.1 | m | n |
| 22 | Sem_060101_TopLevel_012 | Assign and read octetstring with newline character | Clause 6.1.1 | m | n |
| 23 | Sem_060101_TopLevel_013 | Whitespaces, control characters and backslash will be ignored for the octetstring length calculation | Clause 6.1.1 | m | n |
| 24 | Sem_060101_TopLevel_014 | Whitespaces and backslash character is allowed in a universal charstring | Clause 6.1.1 | m | n |
| 25 | Sem_060101_TopLevel_015 | Whitespaces, control characters and backslash will be included for the universal charstring length calculation | Clause 6.1.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 26 | Syn_060101_TopLevel_001 | Assign different bitstring values | Clause 6.1.1 | m | y |
| 27 | Syn_060101_TopLevel_002 | Assign different hexstring values | Clause 6.1.1 | m | y |
| 28 | Syn_060101_TopLevel_003 | Assign different octetstring values | Clause 6.1.1 | m | y |

# 5.18. Accessing individual string elements

*Table 17. Accessing individual string elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010101_AccessStringElements_001 | Accessing not individual elements of a bitstring | Clause 6.1.1.1 | m | y |
| 2 | NegSem_06010101_AccessStringElements_002 | Access bitstring element out of range | Clause 6.1.1.1 | m | y |
| 3 | NegSem_06010101_AccessStringElements_003 | Accessing not individual elements of a hexstring | Clause 6.1.1.1 | m | y |
| 4 | NegSem_06010101_AccessStringElements_004 | Access hexstring element out of range | Clause 6.1.1.1 | m | y |
| 5 | NegSem_06010101_AccessStringElements_005 | Accessing not individual elements of an octetstring | Clause 6.1.1.1 | m | y |
| 6 | NegSem_06010101_AccessStringElements_006 | Access hexstring element out of range | Clause 6.1.1.1 | m | y |
| 7 | Sem_06010101_AccessStringElements_001 | Access bitstring elements | Clause 6.1.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | Sem_06010101_AccessStringElements_002 | Access octetstring elements | Clause 6.1.1.1 | m | y |
| 9 | Sem_06010101_AccessStringElements_003 | Access hexstring elements | Clause 6.1.1.1 | m | y |
| 10 | Sem_06010101_AccessStringElements_004 | Access bitstring elements | Clause 6.1.1.1 | m | y |
| 11 | Sem_06010101_AccessStringElements_005 | Access hexstring elements | Clause 6.1.1.1 | m | y |
| 12 | Sem_06010101_AccessStringElements_006 | Access octetstring elements | Clause 6.1.1.1 | m | y |
| 13 | Sem_06010101_AccessStringElements_007 | Access charstring elements | Clause 6.1.1.1 | m | y |
| 14 | Sem_06010101_AccessStringElements_008 | Access charstring elements | Clause 6.1.1.1 | m | y |
| 15 | Sem_06010101_AccessStringElements_009 | Access charstring elements with nonprintable characters | Clause 6.1.1.1 | m | y |

# 5.19. Lists of values

*Table 18. Lists of values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010201_ListOfValues_001 | Assign values to restricted bitstring. | Clause 6.1.2.1 | m | y |
| 2 | NegSem_06010201_ListOfValues_002 | Assign values to restricted hexstring. | Clause 6.1.2.1 | m | y |
| 3 | NegSem_06010201_ListOfValues_003 | Assign values to restricted octetstring. | Clause 6.1.2.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_06010201_ListOfValues_004 | Assign values to restricted charstring. | Clause 6.1.2.1 | m | y |
| 5 | NegSem_06010201_ListOfValues_005 | Assign values to restricted integer. | Clause 6.1.2.1 | m | y |
| 6 | NegSem_06010201_ListOfValues_006 | Assign values to restricted float. | Clause 6.1.2.1 | m | y |
| 7 | Sem_06010201_ListOfValues_001 | Assign invalid values to restricted bitstring. | Clause 6.1.2.1 | m | y |

# 5.20. Lists of types

*Table 19. Lists of types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010202_ListOfTypes_001 | Assign invalid values to list of types restricted bitstring. | Clause 6.1.2.2 | m | y |
| 2 | NegSem_06010202_ListOfTypes_002 | Assign invalid values to list of types restricted hexstring. | Clause 6.1.2.2 | m | y |
| 3 | NegSem_06010202_ListOfTypes_003 | Assign invalid values to list of types restricted octetstring. | Clause 6.1.2.2 | m | y |
| 4 | NegSem_06010202_ListOfTypes_004 | Assign invalid values to list of types restricted charstring. | Clause 6.1.2.2 | m | y |
| 5 | NegSem_06010202_ListOfTypes_005 | Assign invalid values to list of types restricted universal charstrings. | Clause 6.1.2.2 | m | y |
| 6 | NegSem_06010202_ListOfTypes_006 | Assign invalid values to list of types restricted integers. | Clause 6.1.2.2 | m | y |
| 7 | NegSem_06010202_ListOfTypes_007 | Assign invalid values to list of types restricted floats. | Clause 6.1.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_06010202_ListOfTypes_008 | Assign invalid values to list of types restricted boolean value. | Clause 6.1.2.2 | m | y |
| 9 | NegSem_06010202_ListOfTypes_009 | Assign invalid values to list of types restricted verdicttype. | Clause 6.1.2.2 | m | y |
| 10 | Sem_06010202_ListOfTypes_001 | Assign values to list of types restricted bitstring. | Clause 6.1.2.2 | m | y |
| 11 | Sem_06010202_ListOfTypes_002 | Assign values to list of types restricted hexstring. | Clause 6.1.2.2 | m | y |
| 12 | Sem_06010202_ListOfTypes_003 | Assign values to list of types restricted octetstring. | Clause 6.1.2.2 | m | y |
| 13 | Sem_06010202_ListOfTypes_004 | Assign values to list of types restricted charstring. | Clause 6.1.2.2 | m | y |
| 14 | Sem_06010202_ListOfTypes_005 | Assign values to list of types unicharstring allows non-printable characters | Clause 6.1.2.2 | m | y |
| 15 | Sem_06010202_ListOfTypes_006 | Assign values to list of types restricted integers. | Clause 6.1.2.2 | m | y |
| 16 | Sem_06010202_ListOfTypes_007 | Assign values to list of types restricted floats. | Clause 6.1.2.2 | m | y |
| 17 | Sem_06010202_ListOfTypes_008 | Assign values to list of types restricted boolean value. | Clause 6.1.2.2 | m | y |
| 18 | Sem_06010202_ListOfTypes_009 | Assign values to list of types restricted verdicttype. | Clause 6.1.2.2 | m | y |

## 5.21. Ranges

*Table 20. Ranges*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010203_Ranges_001 | Assign invalid values to restricted integer. | Clause 6.1.2.3 | m | y |
| 2 | NegSem_06010203_Ranges_002 | Assign invalid values to restricted integer. | Clause 6.1.2.3 | m | y |
| 3 | NegSem_06010203_Ranges_003 | Assure that not_a_number is not allowed in float range subtyping. | Clause 6.1.2.3 | m | y |
| 4 | NegSem_06010203_Ranges_004 | Assign invalid values to restricted integer with exclusive bounds. | Clause 6.1.2.3 | m | y |
| 5 | NegSem_06010203_Ranges_005 | Assign invalid values to restricted integer with exclusive bounds. | Clause 6.1.2.3 | m | y |
| 6 | NegSem_06010203_Ranges_006 | Assign range to boolean not permitted. | Clause 6.1.2.3 | m | y |
| 7 | NegSem_06010203_Ranges_007 | Assign invalid value to range constrained charstring. | Clause 6.1.2.3 | m | y |
| 8 | NegSem_06010203_Ranges_008 | Assign invalid value to range constrained charstring. | Clause 6.1.2.3 | m | y |
| 9 | NegSem_06010203_Ranges_009 | Assign invalid value to range constrained charstring. | Clause 6.1.2.3 | m | y |
| 10 | NegSem_06010203_Ranges_010 | Assign invalid values to restricted float. | Clause 6.1.2.3 | m | y |
| 11 | NegSem_06010203_Ranges_011 | Assign invalid values to range restricted float. | Clause 6.1.2.3 | m | y |
| 12 | NegSem_06010203_Ranges_012 | Assign invalid values to range excluded restricted float. | Clause 6.1.2.3 | m | y |
| 13 | NegSem_06010203_Ranges_013 | Assign invalid value to range constrained universal charstring. | Clause 6.1.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | NegSem_06010203_Ranges_014 | Assign invalid value to range constrained universal charstring with mixed bounds. | Clause 6.1.2.3 | m | y |
| 15 | NegSem_06010203_Ranges_015 | Assign invalid value to range constrained charstring. | Clause 6.1.2.3 | m | y |
| 16 | NegSem_06010203_Ranges_016 | Invalid value infinity for range constrained charstring. | Clause 6.1.2.3 | m | y |
| 17 | NegSem_06010203_Ranges_017 | Invalid value -infinity for range constrained charstring. | Clause 6.1.2.3 | m | y |
| 18 | Sem_06010203_Ranges_001 | Assign values to range restricted integer. | Clause 6.1.2.3 | m | y |
| 19 | Sem_06010203_Ranges_002 | Assign values to infinity range restricted integer. | Clause 6.1.2.3 | m | y |
| 20 | Sem_06010203_Ranges_003 | Assign values to range restricted integer with exclusive bounds. | Clause 6.1.2.3 | m | y |
| 21 | Sem_06010203_Ranges_004 | Assign values to range restricted cahrstring with inclusive bounds. | Clause 6.1.2.3 | m | y |
| 22 | Sem_06010203_Ranges_005 | Assign values to range restricted cahrstring with exclusive bounds. | Clause 6.1.2.3 | m | y |
| 23 | Sem_06010203_Ranges_006 | Assign values to range restricted cahrstring with mixed bounds. | Clause 6.1.2.3 | m | y |
| 24 | Sem_06010203_Ranges_007 | Assign values to range restricted universal charstring. | Clause 6.1.2.3 | m | y |
| 25 | Sem_06010203_Ranges_008 | Assign values to range restricted universal charstring with mixed bounds. | Clause 6.1.2.3 | m | y |

## 5.22. String length restrictions

*Table 21. String length restrictions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010204_StringLengthRestrict_001 | Assign invalid values to length restricted bitstring. | Clause 6.1.2.4 | m | y |
| 2 | NegSem_06010204_StringLengthRestrict_002 | Assign invalid values to length restricted bitstring. | Clause 6.1.2.4 | m | y |
| 3 | NegSem_06010204_StringLengthRestrict_003 | Assign invalid values to length restricted hexstring | Clause 6.1.2.4 | m | y |
| 4 | NegSem_06010204_StringLengthRestrict_004 | Assign invalid values to length restricted hexstring | Clause 6.1.2.4 | m | y |
| 5 | NegSem_06010204_StringLengthRestrict_005 | Assign invalid values to length restricted octetstring | Clause 6.1.2.4 | m | y |
| 6 | NegSem_06010204_StringLengthRestrict_006 | Assign invalid values to length restricted octetstring | Clause 6.1.2.4 | m | y |
| 7 | NegSem_06010204_StringLengthRestrict_007 | Assign invalid values to length restricted charstring | Clause 6.1.2.4 | m | y |
| 8 | NegSem_06010204_StringLengthRestrict_008 | Assign invalid values to length restricted charstring | Clause 6.1.2.4 | m | y |
| 9 | NegSyn_06010204_StringLengthRestrict_001 | upper boundary should be greater than lower boundary in string lenght restictions | Clause 6.1.2.4 | m | y |
| 10 | NegSyn_06010204_StringLengthRestrict_002 | boundary integers should be non negative integers | Clause 6.1.2.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_06010204_StringLengthRestrict_001 | Assign values to list of types restricted bitstring. | Clause 6.1.2.4 | m | y |
| 12 | Sem_06010204_StringLengthRestrict_002 | Assign values to list of types restricted hexstring. | Clause 6.1.2.4 | m | y |
| 13 | Sem_06010204_StringLengthRestrict_003 | Assign values to list of types restricted octetstring. | Clause 6.1.2.4 | m | y |
| 14 | Sem_06010204_StringLengthRestrict_004 | Assign values to list of types restricted charstring. | Clause 6.1.2.4 | m | y |

# 5.23. Pattern subtyping of character string types

*Table 22. Pattern subtyping of character string types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06010205_StringPattern_001 | Assign invalid values to pattern restricted character strings. | Clause 6.1.2.5 | m | y |
| 2 | NegSyn_06010205_StringPattern_001 | Assign values to pattern restricted character strings without `@nocase` modifier. | Clause 6.1.2.5 | m | y |
| 3 | NegSyn_06010205_StringPattern_002 | Assign quadruple values to pattern restricted character strings. | Clause 6.1.2.5 | m | y |
| 4 | Sem_06010205_StringPattern_001 | Assign values to pattern restricted character strings. | Clause 6.1.2.5 | m | y |
| 5 | Sem_06010205_StringPattern_002 | Assign values to pattern restricted character strings. | Clause 6.1.2.5 | m | y |
| 6 | Sem_06010205_StringPattern_003 | Assign values to pattern restricted character strings with `@nocase` modifier. | Clause 6.1.2.5 | m | y |

# 5.24. Mixing patterns, lists and ranges

*Table 23. Mixing patterns, lists and ranges*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0601020601_MixingSubtype_001 | Assign invalid values to mixed restricted floats. | Clause 6.1.2.6.1 | m | y |
| 2 | NegSem_0601020601_MixingSubtype_002 | Assign invalid values to mixed restricted integers. | Clause 6.1.2.6.1 | m | y |
| 3 | Sem_0601020601_MixingSubtype_001 | Assign values to mixed restricted floats. | Clause 6.1.2.6.1 | m | y |
| 4 | Sem_0601020601_MixingSubtype_002 | Assign values to mixed restricted integers. | Clause 6.1.2.6.1 | m | y |

# 5.25. Using length restriction with other constraints

*Table 24. Using length restriction with other constraints*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0601020602_StringMixing_001 | Assign invalid values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 2 | NegSem_0601020602_StringMixing_002 | Assign invalid values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 3 | NegSem_0601020602_StringMixing_003 | Assign invalid values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 4 | NegSem_0601020602_StringMixing_004 | Assign invalid values to mixed restricted bit strings. | Clause 6.1.2.6.2 | m | y |
| 5 | NegSem_0601020602_StringMixing_005 | Assign invalid values to mixed restricted hex strings. | Clause 6.1.2.6.2 | m | y |
| 6 | NegSem_0601020602_StringMixing_006 | Assign invalid values to mixed restricted octet strings. | Clause 6.1.2.6.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Sem_0601020602_StringMixing_001 | Assign values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 8 | Sem_0601020602_StringMixing_002 | Assign values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 9 | Sem_0601020602_StringMixing_003 | Assign values to mixed restricted character strings. | Clause 6.1.2.6.2 | m | y |
| 10 | Sem_0601020602_StringMixing_004 | Assign values to mixed restricted bit strings. | Clause 6.1.2.6.2 | m | y |
| 11 | Sem_0601020602_StringMixing_005 | Assign values to mixed restricted hex strings. | Clause 6.1.2.6.2 | m | y |
| 12 | Sem_0601020602_StringMixing_006 | Assign values to mixed restricted octet strings. | Clause 6.1.2.6.2 | m | y |
| 13 | Sem_0601020602_StringMixing_007 | Assign values to pattern restricted character strings using `@nocase` modifier | Clause 6.1.2.6.2 | m | y |

# 5.26. Structured types and values

*Table 25. Structured types and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0602_TopLevel_001 | Value list notation can not be used for a union type. | Clause 6.2 | m | y |
| 2 | NegSem_0602_TopLevel_002 | Indexed notation can not be used for a record type. | Clause 6.2 | m | y |
| 3 | NegSem_0602_TopLevel_003 | Indexed notation can not be used for a set type. | Clause 6.2 | m | y |
| 4 | NegSem_0602_TopLevel_004 | Indexed notation can not be used for a union type. | Clause 6.2 | m | y |
| 5 | NegSyn_0602_TopLevel_001 | Invalid recursive union type definition causing an error | Clause 6.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSyn_0602_TopLevel_002 | Invalid recursive record type definition causing an error | Clause 6.2 | m | y |
| 7 | NegSyn_0602_TopLevel_003 | Combined value list and assignment notation not allowed in the same (immediate) context. | Clause 6.2 | m | y |
| 8 | NegSyn_0602_TopLevel_004 | Combined value list and assignment notation not allowed in the same (immediate) context. | Clause 6.2 | m | y |
| 9 | NegSyn_0602_TopLevel_005 | Combined value list and assignment notation not allowed in the same (immediate) context. | Clause 6.2 | m | y |
| 10 | NegSyn_0602_TopLevel_006 | Combined value list and assignment notation not allowed in the same (immediate) context. | Clause 6.2 | m | y |
| 11 | NegSyn_0602_TopLevel_007 | Combined value list and assignment notation not allowed in the same (immediate) context. | Clause 6.2 | m | y |
| 12 | Sem_0602_TopLevel_001 | Assignment notation can be used for a record type. | Clause 6.2 | m | y |
| 13 | Sem_0602_TopLevel_002 | Assignment notation can be used for a record of type. | Clause 6.2 | m | y |
| 14 | Sem_0602_TopLevel_003 | Assignment notation can be used for a set type. | Clause 6.2 | m | y |
| 15 | Sem_0602_TopLevel_004 | Assignment notation can be used for a set of type. | Clause 6.2 | m | y |
| 16 | Sem_0602_TopLevel_005 | Assignment notation can be used for a union type. | Clause 6.2 | m | y |
| 17 | Sem_0602_TopLevel_006 | Assignment notation can be used for an array. | Clause 6.2 | m | y |
| 18 | Sem_0602_TopLevel_007 | Value list notation can be used for a record type. | Clause 6.2 | m | y |
| 19 | Sem_0602_TopLevel_008 | Value list notation can be used for a record of type. | Clause 6.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 20 | Sem_0602_TopLevel_009 | Indexed notation can be used for an arrays. | Clause 6.2 | m | y |
| 21 | Sem_0602_TopLevel_010 | Value list notation can be used for a set of type. | Clause 6.2 | m | y |
| 22 | Sem_0602_TopLevel_011 | Value list notation can be used for an array. | Clause 6.2 | m | y |
| 23 | Sem_0602_TopLevel_012 | Indexed notation can be used for a record of type. | Clause 6.2 | m | y |
| 24 | Sem_0602_TopLevel_013 | Indexed notation can be used for a set of type. | Clause 6.2 | m | y |
| 25 | Sem_0602_TopLevel_014 | Value list notation can be used for a set type and the values | Clause 6.2 | m | n |
| 26 | Syn_0602_TopLevel_001 | Valid recursive union type definition | Clause 6.2 | m | y |
| 27 | Syn_0602_TopLevel_002 | Valid recursive record type definition | Clause 6.2 | m | y |
| 28 | Syn_0602_TopLevel_003 | Valid recursive record type definition | Clause 6.2 | m | y |
| 29 | Syn_0602_TopLevel_004 | constant definition of a record type. | Clause 6.2 | m | y |
| 30 | Syn_0602_TopLevel_005 | Fields not mentioned are implicitly left unspecified. | Clause 6.2 | m | y |

# 5.27. Record type and values

*Table 26. Record type and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_060201_RecordTypeValues_001 | The omit keyword shall not be used for mandatory fields. | Clause 6.2.1 | m | y |
| 2 | NegSyn_060201_RecordTypeValues_002 | The omit keyword shall not be used for mandatory fields. | Clause 6.2.1 | m | y |
| 3 | Sem_060201_RecordTypeValues_001 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_060201_RecordTypeValues_002 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |
| 5 | Sem_060201_RecordTypeValues_003 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |
| 6 | Syn_060201_RecordTypeValues_001 | The element identifiers are local to the record and shall be unique within the record (but do not have to be globally unique). | Clause 6.2.1 | m | y |
| 7 | Syn_060201_RecordTypeValues_002 | The IUT correctly handles empty record definitions. | Clause 6.2.1 | m | y |
| 8 | NegSyn_060202_SetTypeValues_001 | The omit keyword shall not be used for mandatory fields. | Clause 6.2.1 | m | y |
| 9 | NegSyn_060202_SetTypeValues_002 | The omit keyword shall not be used for mandatory fields. | Clause 6.2.1 | m | y |
| 10 | Sem_060202_SetTypeValues_005 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |
| 11 | Sem_060202_SetTypeValues_006 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |
| 12 | Sem_060202_SetTypeValues_007 | Assignments with `implicit omit` attribute are correctly handled | Clause 6.2.1 | m | y |

# 5.28. Referencing fields of a record type

*Table 27. Referencing fields of a record type*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06020101_ReferencingRecordFields_001 | The dot notation used in record type definitions is correctly handled | Clause 6.2.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_06020101_ReferencingRecordFields_002 | verify that record fields cannot reference themselves | Clause 6.2.1.1 | m | y |
| 3 | NegSem_06020101_ReferencingRecordFields_003 | verify that referencing uninitialized record on the right hand of an assignment is not allowed | Clause 6.2.1.1 | m | y |
| 4 | NegSem_06020101_ReferencingRecordFields_004 | verify that referencing omitted record on the right hand of an assignment is not allowed | Clause 6.2.1.1 | m | y |
| 5 | Sem_06020101_ReferencingRecordFields_001 | The dot notation used in record type definitions is correctly handled | Clause 6.2.1.1 | m | y |
| 6 | Sem_06020101_ReferencingRecordFields_002 | The dot notation used in record type definitions is correctly handled | Clause 6.2.1.1 | m | y |
| 7 | Sem_06020101_ReferencingRecordFields_003 | The dot notation used in record type definitions is correctly handled | Clause 6.2.1.1 | m | y |
| 8 | Sem_06020101_ReferencingRecordFields_004 | The dot notation used in record type definitions is correctly handled | Clause 6.2.1.1 | m | y |
| 9 | Sem_06020101_ReferencingRecordFields_005 | verify that dot notation can be used for referencing elements on the right hand side of an assignement | Clause 6.2.1.1 | m | y |
| 10 | Sem_06020101_ReferencingRecordFields_006 | verify that dot notation can be used for referencing sub-elements on the right hand side of an assignement | Clause 6.2.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_06020101_ReferencingRecordFields_007 | verify that dot notation can be used for referencing function invocation results | Clause 6.2.1.1 | m | y |
| 12 | Sem_06020101_ReferencingRecordFields_008 | verify that mandatory fields are created and uninitialized when expanding uninitialized record values | Clause 6.2.1.1 | m | y |
| 13 | Sem_06020101_ReferencingRecordFields_009 | verify that optional fields are created and uninitialized when expanding uninitialized record values (explicit omit) | Clause 6.2.1.1 | m | y |
| 14 | Sem_06020101_ReferencingRecordFields_010 | verify that optional fields are created and omitted when expanding uninitialized record values (implicit omit) | Clause 6.2.1.1 | m | n |
| 15 | Sem_06020101_ReferencingRecordFields_011 | verify that referencing fields nested deep inside uninitialized record invokes expansion | Clause 6.2.1.1 | m | y |
| 16 | Sem_06020101_ReferencingRecordFields_012 | verify that expansion of uninitialized record values works when other constructive types are involved | Clause 6.2.1.1 | m | y |
| 17 | Sem_06020101_ReferencingRecordFields_013 | verify that mandatory fields are created and uninitialized when expanding omitted record values | Clause 6.2.1.1 | m | y |
| 18 | Sem_06020101_ReferencingRecordFields_014 | verify that optional fields are created and uninitialized when expanding omitted record values (explicit omit) | Clause 6.2.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | Sem_06020101_ReferencingRecordFields_015 | verify that optional fields are created and omitted when expanding omitted record values (implicit omit) | Clause 6.2.1.1 | m | n |
| 20 | Sem_06020101_ReferencingRecordFields_016 | verify that referencing fields nested deep inside omitted record invokes expansion | Clause 6.2.1.1 | m | y |
| 21 | Sem_06020101_ReferencingRecordFields_017 | verify that expansion of omitted record values works when other constructive types are involved | Clause 6.2.1.1 | m | y |

# 5.29. Set type and values

*Table 28. Set type and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060202_SetTypeValues_001 | The dot notation used in set type definitions is correctly handled | Clause 6.2.2 | m | y |
| 2 | Sem_060202_SetTypeValues_001 | The dot notation used in set type definitions is correctly handled | Clause 6.2.2 | m | y |
| 3 | Sem_060202_SetTypeValues_002 | The dot notation used in set type definitions is correctly handled | Clause 6.2.2 | m | y |
| 4 | Sem_060202_SetTypeValues_003 | The dot notation used in set type definitions is correctly handled | Clause 6.2.2 | m | y |
| 5 | Sem_060202_SetTypeValues_004 | The dot notation used in set type definitions is correctly handled | Clause 6.2.2 | m | y |
| 6 | Syn_060202_SetTypeValues_001 | The element identifiers are local to the set and shall be unique within the record (but do not have to be globally unique). | Clause 6.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Syn_060202_SetTypeValues_002 | The IUT correctly handles empty set definitions. | Clause 6.2.2 | m | y |

# 5.30. Records and sets of single types

*Table 29. Records and sets of single types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060203_records_and_sets_of_single_types_003 | negative index applied to a record of value on the right hand side of an assignment | Clause 6.2.3 | m | y |
| 2 | NegSem_060203_records_and_sets_of_single_types_004 | negative index applied to a set of value on the right hand side of an assignment | Clause 6.2.3 | m | y |
| 3 | NegSem_060203_records_and_sets_of_single_types_005 | negative index applied to a record of value on the left hand side of an assignment | Clause 6.2.3 | m | y |
| 4 | NegSem_060203_records_and_sets_of_single_types_006 | negative index applied to a set of value on the left hand side of an assignment | Clause 6.2.3 | m | y |
| 5 | NegSem_060203_records_and_sets_of_single_types_007 | wrong index type applied to a record of value on the right hand side of an assignment | Clause 6.2.3 | m | y |
| 6 | NegSem_060203_records_and_sets_of_single_types_008 | wrong index type applied to a set of value on the right hand side of an assignment | Clause 6.2.3 | m | y |
| 7 | NegSem_060203_records_and_sets_of_single_types_009 | wrong index type applied to a record of value on the left hand side of an assignment | Clause 6.2.3 | m | y |
| 8 | NegSem_060203_records_and_sets_of_single_types_016 | array as a record-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_060203_records_and_sets_of_single_types_017 | array as a record-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 10 | NegSem_060203_records_and_sets_of_single_types_018 | fixed-size record-of as a record-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 11 | NegSem_060203_records_and_sets_of_single_types_019 | fixed-size record-of as a record-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 12 | NegSem_060203_records_and_sets_of_single_types_020 | fixed-size set-of as a record-of value index on right hand side | Clause 6.2.3 | m | y |
| 13 | NegSem_060203_records_and_sets_of_single_types_021 | fixed-size set-of as a record-of value index on left hand side | Clause 6.2.3 | m | y |
| 14 | NegSem_060203_records_and_sets_of_single_types_022 | variable-size record-of as a record-of value index on right hand side | Clause 6.2.3 | m | y |
| 15 | NegSem_060203_records_and_sets_of_single_types_023 | variable-size record-of as a record-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 16 | Sem_060203_records_and_sets_of_single_types_020 | referencing non-existent element of set of value (left-hand side) | Clause 6.2.3 | m | y |
| 17 | Sem_060203_records_and_sets_of_single_types_021 | referencing element of uninitialized record of value (left-hand side) | Clause 6.2.3 | m | y |
| 18 | Sem_060203_records_and_sets_of_single_types_022 | referencing element of uninitialized set of value (left-hand side) | Clause 6.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | Sem_060203_records_and_sets_of_single_types_023 | array as a record-of value index on right hand side (dimensions match) | Clause 6.2.3 | m | y |
| 20 | Sem_060203_records_and_sets_of_single_types_024 | array as a record-of value index on left hand side (dimensions match) | Clause 6.2.3 | m | y |
| 21 | Sem_060203_records_and_sets_of_single_types_025 | array as a record-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 22 | Sem_060203_records_and_sets_of_single_types_026 | array as a record-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 23 | Sem_060203_records_and_sets_of_single_types_027 | fixed-size record-of as a record-of value index on right hand side (dimensions match) | Clause 6.2.3 | m | y |
| 24 | Sem_060203_records_and_sets_of_single_types_028 | fixed-size record-of as a record-of value index on left hand side (dimensions match) | Clause 6.2.3 | m | y |
| 25 | Sem_060203_records_and_sets_of_single_types_029 | fixed-size record-of as a record-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 26 | Sem_060203_records_and_sets_of_single_types_030 | fixed-size record-of as a record-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 27 | Sem_060203_records_and_sets_of_single_types_031 | array as a set-of value index on right hand side (dimensions match) | Clause 6.2.3 | m | y |
| 28 | Sem_060203_records_and_sets_of_single_types_032 | array as a set-of value index on left hand side (dimensions match) | Clause 6.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 29 | Sem_060203_records_and_sets_of_single_types_033 | array as a set-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 30 | Sem_060203_records_and_sets_of_single_types_034 | array as a set-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 31 | Sem_060203_records_and_sets_of_single_types_035 | fixed-size set-of as a record-of value index on right hand side (dimensions match) | Clause 6.2.3 | m | y |
| 32 | Sem_060203_records_and_sets_of_single_types_036 | fixed-size set-of as a record-of value index on left hand side (dimensions match) | Clause 6.2.3 | m | y |
| 33 | Sem_060203_records_and_sets_of_single_types_037 | fixed-size set-of as a record-of value index on right hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |
| 34 | Sem_060203_records_and_sets_of_single_types_038 | fixed-size record-of as a set-of value index on left hand side (less items than record-of dimension) | Clause 6.2.3 | m | y |

# 5.31. Referencing elements of record of and set of types

*Table 30. Referencing elements of record of and set of types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060203_records_and_sets_of_single_types_001 | ensure that the inner type referencing is correctly handled | Clause 6.2.3.2 | m | y |
| 2 | NegSem_060203_records_and_sets_of_single_types_002 | ensure that the inner type referencing is correctly handled | Clause 6.2.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_060203_records_and_sets_of_single_types_010 | wrong index type applied to a set of value on the left hand side of an assignment | Clause 6.2.3.2 | m | y |
| 4 | NegSem_060203_records_and_sets_of_single_types_011 | record of index greater than the upper bound (left-hand side) | Clause 6.2.3.2 | m | n |
| 5 | NegSem_060203_records_and_sets_of_single_types_012 | set of index greater than the upper bound (left-hand side) | Clause 6.2.3.2 | m | n |
| 6 | NegSem_060203_records_and_sets_of_single_types_013 | wrong index type applied to a record of value on the right hand side of an assignment | Clause 6.2.3.2 | m | y |
| 7 | NegSem_060203_records_and_sets_of_single_types_014 | wrong index type applied to a record of value on the right hand side of an assignment | Clause 6.2.3.2 | m | y |
| 8 | NegSem_060203_records_and_sets_of_single_types_015 | verify than an error is generated when sending a partially initialized record of value | Clause 6.2.3.2 | m | y |
| 9 | NegSyn_060203_records_and_sets_of_single_types_001 | ensure that value list cannot contain an empty assignment | Clause 6.2.3.2 | m | y |
| 10 | Sem_060203_records_and_sets_of_single_types_001 | ensure that the inner type referencing is correctly handled | Clause 6.2.3.2 | m | y |
| 11 | Sem_060203_records_and_sets_of_single_types_002 | verify assignment of explicitly identified elements to record of values | Clause 6.2.3.2 | m | n |
| 12 | Sem_060203_records_and_sets_of_single_types_003 | verify assignment of explicitly identified elements to set of values | Clause 6.2.3.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | Sem_060203_records_and_sets_of_single_types_004 | verify handling of missing elements in assignment notation for record of values | Clause 6.2.3.2 | m | y |
| 14 | Sem_060203_records_and_sets_of_single_types_005 | verify handling of missing elements in assignment notation for set of values | Clause 6.2.3.2 | m | y |
| 15 | Sem_060203_records_and_sets_of_single_types_006 | verify handling of missing and ignored elements during record of value re-assignment | Clause 6.2.3.2 | m | n |
| 16 | Sem_060203_records_and_sets_of_single_types_007 | verify handling of missing and ignored elements during record of value re-assignment | Clause 6.2.3.2 | m | n |
| 17 | Sem_060203_records_and_sets_of_single_types_008 | verify handling of value list assignment used for initialization of record of values | Clause 6.2.3.2 | m | y |
| 18 | Sem_060203_records_and_sets_of_single_types_009 | verify handling of value list assignment used for initialization of set of values | Clause 6.2.3.2 | m | y |
| 19 | Sem_060203_records_and_sets_of_single_types_010 | verify handling of value list assignment used for update of record of values | Clause 6.2.3.2 | m | y |
| 20 | Sem_060203_records_and_sets_of_single_types_011 | verify handling of value list assignment used for update of set of values | Clause 6.2.3.2 | m | y |
| 21 | Sem_060203_records_and_sets_of_single_types_012 | verify handling of index notation applied to record of values on right-hand side | Clause 6.2.3.2 | m | y |
| 22 | Sem_060203_records_and_sets_of_single_types_013 | verify handling of index notation applied to set of values on right-hand side | Clause 6.2.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | Sem_060203_records_and_sets_of_single_types_014 | verify handling of index notation applied to record of values on left-hand side | Clause 6.2.3.2 | m | y |
| 24 | Sem_060203_records_and_sets_of_single_types_015 | verify handling of index notation applied to set of values on left-hand side | Clause 6.2.3.2 | m | y |
| 25 | Sem_060203_records_and_sets_of_single_types_016 | verify the first element of a record of value is accessible by an index notation | Clause 6.2.3.2 | m | y |
| 26 | Sem_060203_records_and_sets_of_single_types_017 | verify the first element of a set of value is accessible by an index notation | Clause 6.2.3.2 | m | y |
| 27 | Sem_060203_records_and_sets_of_single_types_019 | referencing non-existent element of record of value (left-hand side) | Clause 6.2.3.2 | m | y |

# 5.32. Enumerated type and values

*Table 31. Enumerated type and values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060204_enumerated_type_and_values_001 | not unique identifiers in enumerated type declaration | Clause 6.2.4 | m | y |
| 2 | NegSem_060204_enumerated_type_and_values_002 | two equal user-assigned enumerated values | Clause 6.2.4 | m | y |
| 3 | NegSem_060204_enumerated_type_and_values_003 | using enumerated value number directly (left hand side of assignments) | Clause 6.2.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_060204_enumerated_type_and_values_004 | using enumerated value number directly (right hand side of assignments) | Clause 6.2.4 | m | y |
| 5 | NegSem_060204_enumerated_type_and_values_005 | using enumerated value without implicit or explicit type reference | Clause 6.2.4 | m | y |
| 6 | NegSem_060204_enumerated_type_and_values_006 | modulepar with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 7 | NegSem_060204_enumerated_type_and_values_007 | formal parameter with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 8 | NegSem_060204_enumerated_type_and_values_008 | constant with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 9 | NegSem_060204_enumerated_type_and_values_009 | variable with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 10 | NegSem_060204_enumerated_type_and_values_010 | template with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 11 | NegSem_060204_enumerated_type_and_values_011 | parameterized template with default parameters and the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | n |
| 12 | NegSem_060204_enumerated_type_and_values_012 | using enumerated value number integer conversion | Clause 6.2.4 | m | y |
| 13 | NegSyn_060204_enumerated_type_and_values_001 | expression as user-assigned enumerated value | Clause 6.2.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_060204_enumerated_type_and_values_001 | reusing enumerated value identifier in another enumerated type declaration | Clause 6.2.4 | m | y |
| 15 | Sem_060204_enumerated_type_and_values_002 | automatic numbering of enumerated items | Clause 6.2.4 | m | y |
| 16 | Sem_060204_enumerated_type_and_values_003 | explicit numbering of enumerated items | Clause 6.2.4 | m | y |
| 17 | Sem_060204_enumerated_type_and_values_004 | mixed automatic and explicit numbering of enumerated items | Clause 6.2.4 | m | y |
| 18 | Sem_060204_enumerated_type_and_values_005 | using enumerated value with implicit type reference | Clause 6.2.4 | m | y |
| 19 | Sem_060204_enumerated_type_and_values_006 | parameterized template without default parameters and with the same name as one of enumerated values of the imported parent type | Clause 6.2.4 | m | y |
| 20 | Sem_060204_enumerated_type_and_values_007 | mixed automatic and explicit numbering of enumerated items | Clause 6.2.4 | m | n |
| 21 | Syn_060204_enumerated_type_and_values_001 | enumerated type declaration | Clause 6.2.4 | m | y |
| 22 | Syn_060204_enumerated_type_and_values_002 | enumerated type declaration with user-assigned values | Clause 6.2.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | Syn_060204_enumerated_type_and_values_003 | constant as user-assigned enumerated values | Clause 6.2.4 | m | y |
| 24 | Syn_060204_enumerated_type_and_values_004 | expression as user-assigned enumerated value | Clause 6.2.4 | m | y |

# 5.33. Unions

*Table 32. Unions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_06020503_nested_type_definition_for_field_types_001 | union type declaration | Clause 6.2.5 | m | y |
| 2 | NegSem_060205_top_level_001 | assignment notation for union values with two items | Clause 6.2.5 | m | y |
| 3 | NegSem_060205_top_level_002 | assignment notation for union values with unknown alternative | Clause 6.2.5 | m | y |
| 4 | NegSem_060205_top_level_003 | ``not used'' symbol in union value notations | Clause 6.2.5 | m | y |
| 5 | NegSem_060205_top_level_004 | omit symbol in union value notations | Clause 6.2.5 | m | y |
| 6 | NegSem_060205_top_level_005 | value list notation used for union value definition | Clause 6.2.5 | m | y |
| 7 | NegSyn_060205_top_level_001 | union type declaration with two equal identifiers | Clause 6.2.5 | m | y |
| 8 | Sem_060205_top_level_001 | assignment notation for union values | Clause 6.2.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | Syn_060205_top_level_001 | union type declaration | Clause 6.2.5 | m | y |
| 10 | Syn_060205_top_level_002 | union type declaration with single item | Clause 6.2.5 | m | y |

# 5.34. Referencing fields of a union type

*Table 33. Referencing fields of a union type*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06020501_referencing_fields_of_union_type_001 | unknown union alternative in value dot notation | Clause 6.2.5.1 | m | y |
| 2 | NegSem_06020501_referencing_fields_of_union_type_002 | unknown union alternative in extended type reference | Clause 6.2.5.1 | m | y |
| 3 | NegSem_06020501_referencing_fields_of_union_type_003 | union alternative referencing itself | Clause 6.2.5.1 | m | y |
| 4 | NegSem_06020501_referencing_fields_of_union_type_004 | union alternative referencing indirectly itself | Clause 6.2.5.1 | m | y |
| 5 | NegSem_06020501_referencing_fields_of_union_type_005 | union alternative costraint passed through extended type reference | Clause 6.2.5.1 | m | y |
| 6 | NegSem_06020501_referencing_fields_of_union_type_006 | referencing not chosen alternative on right hand side of assignment | Clause 6.2.5.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | NegSem_06020501_referencing_fields_of_union_type_007 | referencing alternative of uninitialized union on right hand side of assignment | Clause 6.2.5.1 | m | y |
| 8 | NegSem_06020501_referencing_fields_of_union_type_008 | referencing alternative of omitted union on right hand side of assignment | Clause 6.2.5.1 | m | y |
| 9 | Sem_06020501_referencing_fields_of_union_type_001 | ensure that union is initialized by dot notation | Clause 6.2.5.1 | m | y |
| 10 | Sem_06020501_referencing_fields_of_union_type_002 | union alternative in extended type reference | Clause 6.2.5.1 | m | y |
| 11 | Sem_06020501_referencing_fields_of_union_type_003 | union costraint not applied to extended type reference to its item | Clause 6.2.5.1 | m | y |
| 12 | Sem_06020501_referencing_fields_of_union_type_004 | referencing alternative on left hand side of assignment | Clause 6.2.5.1 | m | y |
| 13 | Sem_06020501_referencing_fields_of_union_type_005 | referencing nested alternative on left hand side of assignment | Clause 6.2.5.1 | m | y |
| 14 | Sem_06020501_referencing_fields_of_union_type_006 | referencing field of structured alternative on left hand side of assignment | Clause 6.2.5.1 | m | y |
| 15 | Sem_06020501_referencing_fields_of_union_type_007 | union is initialized by anytype dot notation | Clause 6.2.5.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 16 | Sem_06020501_referencing_fields_of_union_type_008 | union is initialized by anytype dot notation | Clause 6.2.5.1 | m | y |

# 5.35. Option and union

*Table 34. Option and union*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_06020502_option_and_union_001 | referencing alternative on left hand side of assignment | Clause 6.2.5.2 | m | y |

# 5.36. Anytype

*Table 35. Anytype*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060206_anytype_001 | ensure that after redeclaration of an anytype value the old type and value are lost | Clause 6.2.6 | m | y |
| 2 | NegSem_060206_anytype_002 | Ensure that anytype can not be address type if not explicitly declareted in the module | Clause 6.2.6 | m | y |
| 3 | NegSyn_060206_anytype_001 | ensure that anytype can not be a default type | Clause 6.2.6 | m | n |
| 4 | NegSyn_060206_anytype_002 | ensure that anytype cannot be port type | Clause 6.2.6 | m | y |
| 5 | NegSyn_060206_anytype_003 | ensure that component type not allowed for anytype | Clause 6.2.6 | m | n |
| 6 | Sem_060206_anytype_001 | ensure that anytype comprise integer data type | Clause 6.2.6 | m | y |
| 7 | Sem_060206_anytype_002 | ensure that anytype comprise float data type | Clause 6.2.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | Sem_060206_anytype_003 | ensure that anytype comprise boolean data type | Clause 6.2.6 | m | y |
| 9 | Sem_060206_anytype_004 | ensure that anytype comprise verdicttype data type | Clause 6.2.6 | m | y |
| 10 | Sem_060206_anytype_005 | ensure that anytype comprise bitstring and hexstring data type | Clause 6.2.6 | m | y |
| 11 | Sem_060206_anytype_006 | ensure that ensure that anytype comprise octetstring and charstring | Clause 6.2.6 | m | y |
| 12 | Sem_060206_anytype_007 | ensure that ensure that anytype comprise universal charstring | Clause 6.2.6 | m | y |
| 13 | Sem_060206_anytype_008 | ensure that anytype is a valid value inside an union | Clause 6.2.6 | m | y |
| 14 | Sem_060206_anytype_009 | ensure that record values can be anytype | Clause 6.2.6 | m | y |
| 15 | Sem_060206_anytype_010 | ensure that anytype can be an enum type | Clause 6.2.6 | m | y |
| 16 | Sem_060206_anytype_011 | ensure that anytype can have an set value and set value can be anytype | Clause 6.2.6 | m | y |
| 17 | Sem_060206_anytype_012 | ensure that redeclaration of an anytype value works properly | Clause 6.2.6 | m | y |
| 18 | Sem_060206_anytype_013 | ensure that address type is included to anytype | Clause 6.2.6 | m | y |
| 19 | Sem_060206_anytype_014 | ensure that anytype can be record type | Clause 6.2.6 | m | y |
| 20 | Sem_060206_anytype_015 | ensure that anytype can act as a set type | Clause 6.2.6 | m | y |
| 21 | Sem_060206_anytype_016 | ensure that anytype can act as an union | Clause 6.2.6 | m | y |
| 22 | Sem_060206_anytype_017 | ensure that anytype can comprise array type | Clause 6.2.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | Sem_060206_anytype_018 | ensure that anytype can comprise set of and record of types | Clause 6.2.6 | m | y |
| 24 | Sem_060206_anytype_019 | ensure that anytype can be imported from another module | Clause 6.2.6 | m | y |

## 5.37. Arrays

*Table 36. Arrays*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060207_arrays_001 | ensure that the value limitation is correctly handled within array | Clause 6.2.7 | m | y |
| 2 | NegSem_060207_arrays_002 | ensure that the inner type referencing is correctly handled | Clause 6.2.7 | m | y |
| 3 | NegSem_060207_arrays_003 | negative index applied to an array on the right hand side of an assignment | Clause 6.2.7 | m | y |
| 4 | NegSem_060207_arrays_004 | negative index applied to an array on the left hand side of an assignment | Clause 6.2.7 | m | y |
| 5 | NegSem_060207_arrays_005 | wrong index type applied to an array on the right hand side of an assignment | Clause 6.2.7 | m | y |
| 6 | NegSem_060207_arrays_006 | wrong index type applied to an array on the left hand side of an assignment | Clause 6.2.7 | m | y |
| 7 | NegSem_060207_arrays_007 | array index greater than the upper bound (left-hand side) | Clause 6.2.7 | m | y |
| 8 | NegSem_060207_arrays_008 | wrong index type applied to an array on the right hand side of an assignment | Clause 6.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_060207_arrays_009 | verify than an error is generated when sending a partially initialized array | Clause 6.2.7 | m | y |
| 10 | NegSem_060207_arrays_010 | ensure that the value limitation is correctly handled within array | Clause 6.2.7 | m | y |
| 11 | NegSem_060207_arrays_011 | runtime resolved constant in array type declaration | Clause 6.2.7 | m | y |
| 12 | NegSem_060207_arrays_012 | runtime resolved constant in array variable declaration | Clause 6.2.7 | m | y |
| 13 | NegSem_060207_arrays_013 | variable in array variable declaration | Clause 6.2.7 | m | y |
| 14 | NegSem_060207_arrays_014 | modulepar in array variable declaration | Clause 6.2.7 | m | y |
| 15 | NegSem_060207_arrays_015 | zero dimension array | Clause 6.2.7 | m | y |
| 16 | NegSem_060207_arrays_016 | array with negative dimension | Clause 6.2.7 | m | y |
| 17 | NegSem_060207_arrays_017 | zero in array dimension (range notation) | Clause 6.2.7 | m | n |
| 18 | NegSem_060207_arrays_018 | negative value in array dimension (range notation) | Clause 6.2.7 | m | n |
| 19 | NegSem_060207_arrays_019 | float instead of integer in array dimension | Clause 6.2.7 | m | y |
| 20 | NegSem_060207_arrays_020 | integer array with too many items as multidimensional array index | Clause 6.2.7 | m | y |
| 21 | NegSem_060207_arrays_021 | variable-size record of integer as multidimensional array index | Clause 6.2.7 | m | y |
| 22 | NegSem_060207_arrays_022 | using lower than allowed custom array index on the right hand side of assignments | Clause 6.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | NegSem_060207_arrays_023 | using lower than allowed custom array index on the left hand side of assignments | Clause 6.2.7 | m | y |
| 24 | NegSem_060207_arrays_024 | using greater than allowed custom array index on the right hand side of assignments | Clause 6.2.7 | m | y |
| 25 | NegSem_060207_arrays_025 | using greater than allowed custom array index on the left hand side of assignments | Clause 6.2.7 | m | y |
| 26 | NegSem_060207_arrays_026 | referencing uninitialized array element on the right hand side of assignments | Clause 6.2.7 | m | y |
| 27 | NegSem_060207_arrays_027 | referencing element of uninitialized arrays on the right hand side of assignments | Clause 6.2.7 | m | y |
| 28 | NegSem_060207_arrays_028 | referencing element of omitted arrays on the right hand side of assignments | Clause 6.2.7 | m | y |
| 29 | NegSyn_060207_arrays_001 | ensure that array cannot contain an empty assignment | Clause 6.2.7 | m | y |
| 30 | NegSyn_060207_arrays_002 | ensure that array field cannot contain an empty index | Clause 6.2.7 | m | y |
| 31 | NegSyn_060207_arrays_003 | ensure that array field cannot contain an empty index | Clause 6.2.7 | m | y |
| 32 | NegSyn_060207_arrays_004 | infinity in array variable dimension | Clause 6.2.7 | m | y |
| 33 | NegSyn_060207_arrays_005 | arrays upper value shall not be lesser than the corresponding lower value | Clause 6.2.7 | m | y |
| 34 | Sem_060207_arrays_001 | verify that value list notation can be used for an array | Clause 6.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 35 | Sem_060207_arrays_002 | verify assignment of explicitly identified elements to arrays | Clause 6.2.7 | m | n |
| 36 | Sem_060207_arrays_003 | verify handling of missing elements in assignment notation for arrays | Clause 6.2.7 | m | y |
| 37 | Sem_060207_arrays_004 | verify handling of missing and ignored elements during an array re-assignment | Clause 6.2.7 | m | n |
| 38 | Sem_060207_arrays_005 | verify handling of value list assignment used for initialization of arrays | Clause 6.2.7 | m | y |
| 39 | Sem_060207_arrays_006 | verify handling of value list assignment used for update of arrays | Clause 6.2.7 | m | y |
| 40 | Sem_060207_arrays_007 | verify handling of index notation applied to array on right-hand side | Clause 6.2.7 | m | y |
| 41 | Sem_060207_arrays_008 | verify handling of index notation applied to array on left-hand side | Clause 6.2.7 | m | y |
| 42 | Sem_060207_arrays_009 | verify the first element of an array is accessible by an index notation | Clause 6.2.3.2 | m | y |
| 43 | Sem_060207_arrays_010 | verify that arrays can be used to specify record of type and they are compatible | Clause 6.2.7 | m | y |
| 44 | Sem_060207_arrays_011 | index notation applied to omitted array field on left hand side of assignment | Clause 6.2.7 | m | y |
| 45 | Sem_060207_arrays_012 | referencing element of uninitialized array (left-hand side) | Clause 6.2.7 | m | y |
| 46 | Sem_060207_arrays_013 | ensure that the two dimensional array type referencing is correctly handled | Clause 6.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 47 | Sem_060207_arrays_014 | verify assignment of explicitly identified elements to two dimensional array | Clause 6.2.7 | m | y |
| 48 | Sem_060207_arrays_015 | constant expression in array dimension | Clause 6.2.7 | m | y |
| 49 | Sem_060207_arrays_016 | predefined function in array dimension | Clause 6.2.7 | m | y |
| 50 | Sem_060207_arrays_017 | integer array as multidimensional array index | Clause 6.2.7 | m | y |
| 51 | Sem_060207_arrays_018 | fixed-size record of integer as multidimensional array index | Clause 6.2.7 | m | y |
| 52 | Sem_060207_arrays_019 | integer array as multidimensional array index (less items than dimension count) | Clause 6.2.7 | m | y |
| 53 | Sem_060207_arrays_020 | using custom array index on the right hand side of assignments | Clause 6.2.7 | m | y |
| 54 | Sem_060207_arrays_021 | using custom array index on the left hand side of assignments | Clause 6.2.7 | m | y |
| 55 | Sem_060207_arrays_022 | using less indexes than array dimensions on the right hand side of assignments | Clause 6.2.7 | m | y |
| 56 | Sem_060207_arrays_023 | using less indexes than array dimensions on the left hand side of assignments | Clause 6.2.7 | m | y |
| 57 | Syn_060207_arrays_001 | array specified in variable declaration | Clause 6.2.7 | m | y |
| 58 | Syn_060207_arrays_002 | multidimensional array type declaration | Clause 6.2.7 | m | y |
| 59 | Syn_060207_arrays_003 | multidimensional array specified in variable declaration | Clause 6.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 60 | Syn_060207_arrays_004 | array type dimension specified as a range | Clause 6.2.7 | m | y |
| 61 | Syn_060207_arrays_005 | multiple array type dimensions specified as a range | Clause 6.2.7 | m | y |
| 62 | Syn_060207_arrays_006 | array variable dimension specified as a range | Clause 6.2.7 | m | y |
| 63 | Syn_060207_arrays_007 | multiple array variable dimensions specified as a range | Clause 6.2.7 | m | y |

## 5.38. The default type

*Table 37. The default type*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_060208_default_type_001 | verify than a reference to an activated default can be assigned to a default variable | Clause 6.2.8 | m | y |
| 2 | Sem_060208_default_type_002 | verify than `null` value can be assigned to a default variable | Clause 6.2.8 | m | y |
| 3 | Sem_060208_default_type_003 | verify than existing default references can be assigned | Clause 6.2.8 | m | y |

## 5.39. Communication port types

*Table 38. Communication port types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060209_CommunicationPortTypes_001 | Restriction of port definitions are appropriately handles | Clause 6.2.9 | m | n |
| 2 | NegSem_060209_CommunicationPortTypes_002 | Restriction of port definitions are appropriately handles | Clause 6.2.9 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_060209_CommunicationPortTypes_003 | Restriction of port definitions are appropriately handles | Clause 6.2.9 | m | n |
| 4 | NegSem_060209_CommunicationPortTypes_004 | Verify that an error is generated when a message port type definition contains no message types | Clause 6.2.9 | m | y |
| 5 | NegSem_060209_CommunicationPortTypes_005 | Verify that an error is generated when a procedure port type definition contains no signatures | Clause 6.2.9 | m | y |
| 6 | NegSem_060209_CommunicationPortTypes_006 | Verify that an error is generated when a signature port definition contains multiple `address` clauses | Clause 6.2.9 | m | n |
| 7 | NegSem_060209_CommunicationPortTypes_007 | Verify that an error is generated when a signature port definition contains multiple `map` clauses | Clause 6.2.9 | m | n |
| 8 | NegSem_060209_CommunicationPortTypes_008 | Verify that an error is generated when a signature port definition contains multiple `unmap` clauses | Clause 6.2.9 | m | n |
| 9 | Sem_060209_CommunicationPortTypes_004 | `Map` and `unmap` param and local port address are allowed in a testcase block | Clause 6.2.9 | m | n |
| 10 | Sem_060209_CommunicationPortTypes_005 | Parameter `MessageType` of the port shall be data type | Clause 6.2.9 | m | n |
| 11 | Syn_060209_CommunicationPortTypes_001 | Message-based ports are accepted. | Clause 6.2.9 | m | y |
| 12 | Syn_060209_CommunicationPortTypes_002 | Message-based ports with address are accepted. | Clause 6.2.9 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | Syn_060209_Communication PortTypes_003 | Verify that it is possible to define procedute-based port types | Clause 6.2.9 | m | y |
| 14 | Syn_060209_Communication PortTypes_004 | Procedure-based ports with address are accepted | Clause 6.2.9 | m | n |
| 15 | Syn_060209_Communication PortTypes_005 | `Map` param is accepted by the port definition. | Clause 6.2.9 | m | n |
| 16 | Syn_060209_Communication PortTypes_006 | `Unmap` param is accepted by the port definition. | Clause 6.2.9 | m | n |
| 17 | Syn_060209_Communication PortTypes_007 | Complex port definition are accepted. | Clause 6.2.9 | m | n |
| 18 | Syn_060209_Communication PortTypes_008 | Procedure-base port type definition can contain `map` parameter definition | Clause 6.2.9 | m | n |
| 19 | Syn_060209_Communication PortTypes_009 | Procedure-base port type definition can contain `unmap` parameter definition | Clause 6.2.9 | m | n |
| 20 | Syn_060209_Communication PortTypes_010 | Complex procedure-based port type definition are accepted | Clause 6.2.9 | m | n |

# 5.40. Component types

*Table 39. Component types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_060210_ReuseofComponentTypes_001 | Cyclic extension is not allowed | Clause 6.2.10 | m | y |
| 2 | NegSyn_060210_ReuseofComponentTypes_002 | Extending a component that occurs name clash is not allowed | Clause 6.2.10 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSyn_060210_ReuseofComponentTypes_003 | Extending a component that occurs name clash is not allowed | Clause 6.2.10 | m | y |
| 4 | Sem_060210_ReuseofComponentTypes_001 | Extending a component with another component works properly | Clause 6.2.10 | m | y |
| 5 | Sem_060210_ReuseofComponentTypes_002 | Extending a component with several other component works properly | Clause 6.2.10 | m | y |
| 6 | Sem_060210_ReuseofComponentTypes_003 | Extending a component with and extended component works properly | Clause 6.2.10 | m | y |

# 5.41. Addressing entities inside the SUT

*Table 40. Addressing entities inside the SUT*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060212_AddressingEntitiesInsideSut_001 | Ensure right type checking for address types in ports | Clause 6.2.12 | m | n |
| 2 | NegSem_060212_AddressingEntitiesInsideSut_002 | Address type cannot be used in a from part of `receive` operation with connected ports | Clause 6.2.12 | m | n |
| 3 | NegSem_060212_AddressingEntitiesInsideSut_003 | Address type cannot be used in a sender part of `receive` operation with connected ports | Clause 6.2.12 | m | n |
| 4 | NegSem_060212_AddressingEntitiesInsideSut_004 | Address type cannot be used in a to part of `sender` operation with connected ports | Clause 6.2.12 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | Sem_060212_AddressingEntitiesInsideSut_001 | Ensure `null` assignment is accepted for addresses | Clause 6.2.12 | m | n |
| 6 | Sem_060212_AddressingEntitiesInsideSut_002 | The right port address is used | Clause 6.2.12 | m | n |

## 5.42. Subtyping of structured types

*Table 41. Subtyping of structured types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_06021301_LengthSubtyping_001 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 2 | NegSem_06021301_LengthSubtyping_002 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 3 | NegSem_06021301_LengthSubtyping_003 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 4 | NegSem_06021301_LengthSubtyping_004 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 5 | NegSem_06021301_LengthSubtyping_005 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 6 | NegSem_06021301_LengthSubtyping_006 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 7 | Syn_06021301_LengthSubtyping_001 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |
| 8 | Syn_06021301_LengthSubtyping_002 | The length subtyping check for `record of' or `set of' types | Clause 6.2.13.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_06021302_ListSubtyping_001 | ensure that list subtyping check for record types is properly handled | Clause 6.2.13.2 | m | y |
| 10 | NegSem_06021302_ListSubtyping_002 | ensure that list subtyping check for record types is properly handled | Clause 6.2.13.2 | m | y |
| 11 | Sem_06021302_ListSubtyping_001 | ensure that list subtyping check for record types is properly handled | Clause 6.2.13.2 | m | y |
| 12 | Sem_06021302_ListSubtyping_002 | ensure that list subtyping check for record types is properly handled | Clause 6.2.13.2 | m | n |
| 13 | Sem_06021302_ListSubtyping_003 | ensure that list subtyping check for record types is properly handled | Clause 6.2.13.2 | m | n |

# 5.43. Type compatibility of non-structured types

*Table 42. Type compatibility of non-structured types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060301_non_structured_types_001 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |
| 2 | NegSem_060301_non_structured_types_002 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |
| 3 | NegSem_060301_non_structured_types_003 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |
| 4 | NegSem_060301_non_structured_types_004 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSem_060301_non_structured_types_005 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |
| 6 | NegSem_060301_non_structured_types_006 | The IUT correctly handles assignments from incompatible type ranges | Clause 6.3.1 | m | n |
| 7 | NegSem_060301_non_structured_types_007 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | y |
| 8 | NegSem_060301_non_structured_types_008 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | y |
| 9 | NegSem_060301_non_structured_types_009 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | n |
| 10 | NegSem_060301_non_structured_types_010 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | n |
| 11 | NegSem_060301_non_structured_types_011 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | n |
| 12 | NegSem_060301_non_structured_types_012 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | n |
| 13 | Sem_060301_non_structured_types_001 | The IUT correctly handles assignments from compatible type ranges | Clause 6.3.1 | m | y |
| 14 | Sem_060301_non_structured_types_002 | The IUT correctly handles assignments from compatible size restrictions | Clause 6.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 15 | Sem_060301_non_structured_types_003 | The IUT correctly handles assignments from compatible type ranges | Clause 6.3.1 | m | y |
| 16 | Sem_060301_non_structured_types_004 | The IUT correctly handles assignments from compatible type ranges | Clause 6.3.1 | m | y |

# 5.44. Type compatibility of structured types

*Table 43. Type compatibility of structured types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060302_structured_types_002 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | y |
| 2 | NegSem_060302_structured_types_003 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | n |
| 3 | NegSem_060302_structured_types_004 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | y |
| 4 | NegSem_060302_structured_types_005 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | n |
| 5 | NegSem_060302_structured_types_006 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | n |
| 6 | NegSem_060302_structured_types_007 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | n |
| 7 | NegSem_060302_structured_types_008 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_060302_structured_types_009 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | y |
| 9 | NegSem_060302_structured_types_010 | The IUT rejects assignments from incompatible types or type ranges | Clause 6.3.2 | m | y |
| 10 | NegSem_060302_structured_types_011 | The IUT rejects assignments from structures having incompatible anytypes | Clause 6.3.2 | m | y |
| 11 | NegSem_060302_structured_types_012 | The IUT rejects assignments having mismatch between undefined and omitted elements | Clause 6.3.2 | m | n |
| 12 | NegSem_060302_structured_types_013 | The IUT rejects assignments having mismatch between undefined and omitted elements | Clause 6.3.2 | m | n |
| 13 | NegSem_060302_structured_types_014 | The IUT rejects assignments between incompatible structures | Clause 6.3.2 | m | n |
| 14 | NegSem_060302_structured_types_015 | The IUT rejects assignments between incompatible structures | Clause 6.3.2 | m | n |
| 15 | NegSem_060302_structured_types_016 | The IUT rejects assignments between incompatible structures | Clause 6.3.2 | m | y |
| 16 | NegSem_060302_structured_types_017 | The IUT rejects assignments between incompatible structures | Clause 6.3.2 | m | n |
| 17 | NegSem_060302_structured_types_018 | The IUT rejects assignments between incompatible structures | Clause 6.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | NegSem_060302_structured_types_019 | The IUT correctly handles assignments from structures having compatible types and lengths | Clause 6.3.2 | m | n |
| 19 | Sem_060302_structured_types_001 | The IUT correctly handles assignments from structures having compatible types and type ranges | Clause 6.3.2 | m | y |
| 20 | Sem_060302_structured_types_002 | The IUT correctly handles assignments from structures having compatible types and lengths | Clause 6.3.2 | m | y |
| 21 | Sem_060302_structured_types_003 | The IUT correctly handles assignments from structures having compatible types and type ranges | Clause 6.3.2 | m | y |
| 22 | Sem_060302_structured_types_004 | The IUT correctly handles assignments from structures having compatible anytypes | Clause 6.3.2 | m | y |
| 23 | Sem_060302_structured_types_005 | The IUT correctly handles assignments from structures having compatible types and type ranges | Clause 6.3.2 | m | y |
| 24 | Sem_060302_structured_types_006 | The IUT correctly handles assignments from structures having compatible types and lengths | Clause 6.3.2 | m | n |

# 5.45. Type compatibility of enumerated types

*Table 44. Type compatibility of enumerated types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060302_structured_types_001 | Reject assignment of other enumerated types since they are only compatible to synonym types | Clause 6.3.2.1 | m | y |

# 5.46. Type compatibility of component types

*Table 45. Type compatibility of component types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060303_component_types_001 | The IUT correctly handles component incompatibility due to differing list of constant definitions | Clause 6.3.3 | m | y |
| 2 | NegSem_060303_component_types_002 | The IUT correctly handles component incompatibility due to differing constant types having same name | Clause 6.3.3 | m | y |
| 3 | NegSem_060303_component_types_003 | Ensure that the IUT correctly handles component compatibility of different `runs on` clauses | Clause 6.3.3 | m | y |
| 4 | NegSem_060303_component_types_004 | Ensure that the IUT correctly handles component compatibility of `mtc` and `runs on` clause | Clause 6.3.3 | m | y |
| 5 | NegSem_060303_component_types_005 | Ensure that the IUT correctly handles component compatibility of `system` and `runs on` clause | Clause 6.3.3 | m | y |
| 6 | NegSem_060303_component_types_006 | Ensure that the IUT correctly handles component compatibility of different `system` clauses | Clause 6.3.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Sem_060303_component_types_001 | The IUT correctly handles assignments from structures having compatible components | Clause 6.3.3 | m | y |
| 8 | Sem_060303_component_types_002 | The IUT correctly handles assignments from structures having compatible components | Clause 6.3.3 | m | y |
| 9 | Sem_060303_component_types_003 | Ensure that the IUT correctly handles component compatibility of different `runs on` clauses | Clause 6.3.3 | m | n |
| 10 | Sem_060303_component_types_004 | Ensure that the IUT correctly handles component compatibility of `mtc` and `runs on` clause | Clause 6.3.3 | m | y |
| 11 | Sem_060303_component_types_005 | Ensure that the IUT correctly handles component compatibility of `system` and `runs on` clause | Clause 6.3.3 | m | y |
| 12 | Sem_060303_component_types_006 | Ensure that the IUT correctly handles component compatibility of different `system` clauses | Clause 6.3.3 | m | y |

# 5.47. Type compatibility of communication operations

*Table 46. Type compatibility of communication operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_060304_compatibility_of_communication_operations_001 | compatible but not strongly typed value in `send` operation | Clause 6.3.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_060304_compatibility_of_communication_operations_002 | compatible but not strongly typed value in `receive` operation | Clause 6.3.4 | m | n |
| 3 | NegSem_060304_compatibility_of_communication_operations_003 | compatible but not strongly typed value in `raise` operation | Clause 6.3.4 | m | n |
| 4 | NegSem_060304_compatibility_of_communication_operations_004 | compatible but not strongly typed value in `raise` operation | Clause 6.3.4 | m | n |
| 5 | NegSem_060304_compatibility_of_communication_operations_005 | compatible but not strongly typed value in `trigger` operation | Clause 6.3.4 | m | n |

## 5.48. Expression

*Table 47. Expression*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_07_toplevel_001 | function without `return` clause in expression | Clause 7 | m | y |
| 2 | NegSem_07_toplevel_002 | template used as expression operand | Clause 7 | m | y |
| 3 | NegSem_07_toplevel_003 | uninitialized value in an expression | Clause 7 | m | y |
| 4 | NegSem_07_toplevel_004 | partially initialized value in an expression | Clause 7 | m | n |
| 5 | NegSem_07_toplevel_005 | null value in an expression | Clause 7 | m | n |
| 6 | Sem_07_toplevel_001 | expression composed of several expressions | Clause 7 | m | y |
| 7 | Sem_07_toplevel_002 | compound expression as an operand of array type | Clause 7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | Sem_07_toplevel_003 | compound expression as an operand of record type | Clause 7 | m | y |
| 9 | Sem_07_toplevel_004 | compound expression as an operand of record-of type | Clause 7 | m | y |
| 10 | Sem_07_toplevel_005 | compound expression as an operand of set-of type | Clause 7 | m | y |
| 11 | Sem_07_toplevel_006 | element of partially initialized structured value | Clause 7 | m | y |
| 12 | Sem_07_toplevel_007 | compound expression as an operand of set-of type | Clause 7 | m | y |
| 13 | Sem_07_toplevel_008 | compound expression as an operand of set-of type | Clause 7 | m | y |
| 14 | Sem_07_toplevel_009 | compound expression as an operand of set type | Clause 7 | m | y |

# 5.49. Arithmetic operators

*Table 48. Arithmetic operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_070101_ArithmeticOperators_001 | Arithmetic operators are for integer and float values | Clause 7.1.1 | m | y |
| 2 | NegSem_070101_ArithmeticOperators_002 | Arithmetic operators can handle same type of variables | Clause 7.1.1 | m | y |
| 3 | NegSem_070101_ArithmeticOperators_003 | Mod arithmetic operator can handle integer variables | Clause 7.1.1 | m | y |
| 4 | NegSem_070101_ArithmeticOperators_004 | Rem arithmetic operator can handle integer variables | Clause 7.1.1 | m | y |
| 5 | NegSem_070101_ArithmeticOperators_008 | In x mod y arithmetic operator y is non-zero positive number | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_070101_ArithmeticOperators_009 | In x rem y arithmetic operator y is non-zero positive number | Clause 7.1.1 | m | y |
| 7 | NegSem_070101_ArithmeticOperators_010 | In x rem y arithmetic operator y is non-zero positive number | Clause 7.1.1 | m | y |
| 8 | Sem_070101_ArithmeticOperators_001 | The addition of two integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 9 | Sem_070101_ArithmeticOperators_002 | The addition of multiple integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 10 | Sem_070101_ArithmeticOperators_003 | The addition of two integer variables is evaluated correctly when the expression contains a negative value. | Clause 7.1.1 | m | y |
| 11 | Sem_070101_ArithmeticOperators_004 | The substraction of two integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 12 | Sem_070101_ArithmeticOperators_005 | The substraction of multiple integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 13 | Sem_070101_ArithmeticOperators_006 | The multiplication of two integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 14 | Sem_070101_ArithmeticOperators_007 | The multiplication of multiple integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 15 | Sem_070101_ArithmeticOperators_008 | The division of two integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 16 | Sem_070101_ArithmeticOperators_009 | The division of multiple integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 17 | Sem_070101_ArithmeticOperators_010 | The application of the modulo operator on integer variables is evaluated correctly when the remainder is zero. | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | Sem_070101_ArithmeticOperators_011 | The application of the modulo operator on integer variables is evaluated correctly when the integer value is smaller than the modulo value. | Clause 7.1.1 | m | y |
| 19 | Sem_070101_ArithmeticOperators_012 | The application of the modulo operator on integer variables is evaluated correctly when the integer value greater than the modulo value. | Clause 7.1.1 | m | y |
| 20 | Sem_070101_ArithmeticOperators_013 | The application of the modulo operator on integer variables is evaluated correctly when two consecutive modulo operators are applied. | Clause 7.1.1 | m | y |
| 21 | Sem_070101_ArithmeticOperators_014 | The application of the modulo operator on integer variables is evaluated correctly when the operand is a negative integer. | Clause 7.1.1 | m | y |
| 22 | Sem_070101_ArithmeticOperators_015 | The application of the remainder operator on integer variables is evaluated correctly when the operand is a negative integer. | Clause 7.1.1 | m | y |
| 23 | Sem_070101_ArithmeticOperators_016 | The application of the remainder operator on integer variables is evaluated correctly when the operand is a negative integer. | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 24 | Sem_070101_ArithmeticOperators_017 | The consecutive application of the remainder operator and the modulo operator on integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 25 | Sem_070101_ArithmeticOperators_018 | Operator combinations and the modulo operator on integer variables is evaluated correctly. | Clause 7.1.1 | m | y |
| 26 | Sem_070101_ArithmeticOperators_019 | The addition operator works on float variables. | Clause 7.1.1 | m | y |
| 27 | Sem_070101_ArithmeticOperators_020 | The substraction operator works on float variables. | Clause 7.1.1 | m | y |
| 28 | Sem_070101_ArithmeticOperators_021 | The multiplication operator works on float variables. | Clause 7.1.1 | m | y |
| 29 | Sem_070101_ArithmeticOperators_022 | The division operator works on float variables. | Clause 7.1.1 | m | y |
| 30 | Sem_070101_ArithmeticOperators_023 | The combination of different operators works on float variables. | Clause 7.1.1 | m | y |
| 31 | Sem_070101_ArithmeticOperators_024 | The operator precedence is evaluated correctly. | Clause 7.1.1 | m | y |
| 32 | Sem_070101_ArithmeticOperators_025 | The operator precedence is evaluated correctly. | Clause 7.1.1 | m | y |
| 33 | Sem_070101_ArithmeticOperators_026 | The operator precedence is evaluated correctly. | Clause 7.1.1 | m | y |
| 34 | Sem_070101_ArithmeticOperators_027 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 35 | Sem_070101_ArithmeticOperators_028 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 36 | Sem_070101_ArithmeticOperators_029 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 37 | Sem_070101_ArithmeticOperators_030 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 38 | Sem_070101_ArithmeticOperators_031 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 39 | Sem_070101_ArithmeticOperators_032 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 40 | Sem_070101_ArithmeticOperators_033 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 41 | Sem_070101_ArithmeticOperators_034 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 42 | Sem_070101_ArithmeticOperators_035 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 43 | Sem_070101_ArithmeticOperators_036 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 44 | Sem_070101_ArithmeticOperators_037 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 45 | Sem_070101_ArithmeticOperators_038 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 46 | Sem_070101_ArithmeticOperators_039 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 47 | Sem_070101_ArithmeticOperators_040 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 48 | Sem_070101_ArithmeticOperators_041 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 49 | Sem_070101_ArithmeticOperators_042 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 50 | Sem_070101_ArithmeticOperators_043 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 51 | Sem_070101_ArithmeticOperators_044 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 52 | Sem_070101_ArithmeticOperators_045 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 53 | Sem_070101_ArithmeticOperators_046 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 54 | Sem_070101_ArithmeticOperators_047 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 55 | Sem_070101_ArithmeticOperators_048 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 56 | Sem_070101_ArithmeticOperators_049 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 57 | Sem_070101_ArithmeticOperators_050 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 58 | Sem_070101_ArithmeticOperators_051 | Arithmetic operators can handle special float values | Clause 7.1.1 | m | y |
| 59 | Syn_070101_ArithmeticOperators_001 | The addition of two integers in a constant is accepted. | Clause 7.1.1 | m | y |
| 60 | Syn_070101_ArithmeticOperators_002 | The substraction of two integers in a constant is accepted. | Clause 7.1.1 | m | y |
| 61 | Syn_070101_ArithmeticOperators_003 | The multiplication of two integers in a constant is accepted. | Clause 7.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 62 | Syn_070101_ArithmeticOperators_004 | The division of two integers in a constant is accepted. | Clause 7.1.1 | m | y |
| 63 | Syn_070101_ArithmeticOperators_005 | The modulo operator on two integers is accepted. | Clause 7.1.1 | m | y |
| 64 | Syn_070101_ArithmeticOperators_006 | The remainder operator on two integers is accepted. | Clause 7.1.1 | m | y |
| 65 | Syn_070101_ArithmeticOperators_007 | Operator combinations on integers is accepted. | Clause 7.1.1 | m | y |
| 66 | Syn_070101_ArithmeticOperators_008 | The addition operator on float constants is accepted. | Clause 7.1.1 | m | y |
| 67 | Syn_070101_ArithmeticOperators_009 | The substraction operator on float constants is accepted. | Clause 7.1.1 | m | y |
| 68 | Syn_070101_ArithmeticOperators_010 | The multiplication operator on float constants is accepted. | Clause 7.1.1 | m | y |
| 69 | Syn_070101_ArithmeticOperators_011 | The division operator on float constants is accepted. | Clause 7.1.1 | m | y |
| 70 | Syn_070101_ArithmeticOperators_012 | A combination of operators on float constants is accepted. | Clause 7.1.1 | m | y |

# 5.50. List operator

*Table 49. List operator*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070102_ListOperator_001 | The list operator on bitstrings is evaluated correctly. | Clause 7.1.2 | m | y |
| 2 | Sem_070102_ListOperator_002 | The list operator on charstrings is evaluated correctly. | Clause 7.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | Sem_070102_ListOperator_003 | The list operator on record of is evaluated correctly. | Clause 7.1.2 | m | y |
| 4 | Sem_070102_ListOperator_004 | The list operator on set of is evaluated correctly. | Clause 7.1.2 | m | y |
| 5 | Sem_070102_ListOperator_005 | The list operator on arrays is evaluated correctly. | Clause 7.1.2 | m | n |
| 6 | Sem_070102_ListOperator_006 | The list operator on record of is evaluated correctly. | Clause 7.1.2 | m | y |

# 5.51. Relational operators

*Table 50. Relational operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070101_ArithmeticOperators_051 | The equal to operator on address with value `null` is evaulated correctly | Clause 7.1.3 | m | n |
| 2 | Sem_070101_ArithmeticOperators_052 | The not equal to operator on address with value `null` is evaulated correctly | Clause 7.1.3 | m | n |
| 3 | NegSem_070103_RelationalOperators_001 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | n |
| 4 | NegSem_070103_RelationalOperators_002 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 5 | NegSem_070103_RelationalOperators_003 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 6 | NegSem_070103_RelationalOperators_004 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 7 | NegSem_070103_RelationalOperators_005 | The not equal to operator on address can not be evaluated if value is uninitialized. | Clause 7.1.3 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSyn_070103_RelationalOperators_001 | The greater operator on address can not be evaluated. | Clause 7.1.3 | m | n |
| 9 | NegSyn_070103_RelationalOperators_002 | The less operator on address can not be evaluated. | Clause 7.1.3 | m | n |
| 10 | NegSyn_070103_RelationalOperators_003 | The less or equal to operator on address can not be evaluated. | Clause 7.1.3 | m | n |
| 11 | NegSyn_070103_RelationalOperators_004 | The greater or equal to operator on address can not be evaluated. | Clause 7.1.3 | m | n |
| 12 | Sem_070103_RelationalOperators_001 | The equals operator on integers is evaluated correctly. | Clause 7.1.3 | m | y |
| 13 | Sem_070103_RelationalOperators_002 | The equals operator on floats is evaluated correctly. | Clause 7.1.3 | m | y |
| 14 | Sem_070103_RelationalOperators_003 | The equals operator on enumerations is evaluated correctly. | Clause 7.1.3 | m | y |
| 15 | Sem_070103_RelationalOperators_004 | The less than operator on integers is evaluated correctly. | Clause 7.1.3 | m | y |
| 16 | Sem_070103_RelationalOperators_005 | The less than operator on floats is evaluated correctly. | Clause 7.1.3 | m | y |
| 17 | Sem_070103_RelationalOperators_006 | The less than operator on enumerations is evaluated correctly. | Clause 7.1.3 | m | y |
| 18 | Sem_070103_RelationalOperators_007 | The less than or equal to operator on integers is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 19 | Sem_070103_RelationalOperators_008 | The less than or equal to operator on integers is evaluated correctly with equal values. | Clause 7.1.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 20 | Sem_070103_RelationalOperators_009 | The less than or equal to operator on floats is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 21 | Sem_070103_RelationalOperators_010 | The less than or equal to operator on floats is evaluated correctly with equal values. | Clause 7.1.3 | m | y |
| 22 | Sem_070103_RelationalOperators_011 | The less than or equal to operator on enumerations is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 23 | Sem_070103_RelationalOperators_012 | The less than or equal to operator on enumerations is evaluated correctly with equal values. | Clause 7.1.3 | m | y |
| 24 | Sem_070103_RelationalOperators_013 | The greater than operator on integers is evaluated correctly. | Clause 7.1.3 | m | y |
| 25 | Sem_070103_RelationalOperators_014 | The less than operator on floats is evaluated correctly. | Clause 7.1.3 | m | y |
| 26 | Sem_070103_RelationalOperators_015 | The less than operator on enumerations is evaluated correctly. | Clause 7.1.3 | m | y |
| 27 | Sem_070103_RelationalOperators_016 | The greater than or equal to operator on integers is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 28 | Sem_070103_RelationalOperators_017 | The greater than or equal to operator on integers is evaluated correctly with equal values. | Clause 7.1.3 | m | y |
| 29 | Sem_070103_RelationalOperators_018 | The greater than or equal to operator on floats is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 30 | Sem_070103_RelationalOperators_019 | The greater than or equal to operator on floats is evaluated correctly with equal values. | Clause 7.1.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 31 | Sem_070103_RelationalOperators_020 | The less than or equal to operator on enumerations is evaluated correctly with differing values. | Clause 7.1.3 | m | y |
| 32 | Sem_070103_RelationalOperators_021 | The greater than or equal to operator on enumerations is evaluated correctly with equal values. | Clause 7.1.3 | m | y |
| 33 | Sem_070103_RelationalOperators_022 | The not equals operator on integers is evaluated correctly. | Clause 7.1.3 | m | y |
| 34 | Sem_070103_RelationalOperators_023 | The not equals operator on floats is evaluated correctly. | Clause 7.1.3 | m | y |
| 35 | Sem_070103_RelationalOperators_024 | The not equals operator on enumerations is evaluated correctly. | Clause 7.1.3 | m | y |
| 36 | Sem_070103_RelationalOperators_025 | The equals operator on sets is evaluated correctly. | Clause 7.1.3 | m | y |
| 37 | Sem_070103_RelationalOperators_026 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 38 | Sem_070103_RelationalOperators_030 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 39 | Sem_070103_RelationalOperators_031 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 40 | Sem_070103_RelationalOperators_032 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 41 | Sem_070103_RelationalOperators_033 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |
| 42 | Sem_070103_RelationalOperators_034 | The equals operator on records is evaluated correctly. | Clause 7.1.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 43 | Sem_070103_RelationalOperators_035 | The eqaul to operator on address is evaluated correctly with equal values. | Clause 7.1.3 | m | n |
| 44 | Sem_070103_RelationalOperators_036 | The eqaul to operator on address is evaluated correctly with equal values. | Clause 7.1.3 | m | y |
| 45 | Sem_070103_RelationalOperators_037 | The not eqaul to operator on record type address is evaluated correctly. | Clause 7.1.3 | m | n |
| 46 | Sem_070103_RelationalOperators_038 | Less than operator evaulates correctly infinity special float | Clause 7.1.3 | m | y |
| 47 | Sem_070103_RelationalOperators_039 | Less than or equal to operator evaulates correctly infinity special float | Clause 7.1.3 | m | y |
| 48 | Sem_070103_RelationalOperators_040 | Greather than operator evaulates correctly -infinity special float | Clause 7.1.3 | m | y |
| 49 | Sem_070103_RelationalOperators_041 | Greather than or equal to operator evaulates correctly -infinity special float | Clause 7.1.3 | m | y |
| 50 | Sem_070103_RelationalOperators_042 | Equal to operator evaulates correctly -infinity special float | Clause 7.1.3 | m | y |
| 51 | Sem_070103_RelationalOperators_043 | Equal to operator evaulates correctly infinity special float | Clause 7.1.3 | m | y |
| 52 | Sem_070103_RelationalOperators_044 | Not equal to operator evaulates correctly infinity special float | Clause 7.1.3 | m | y |
| 53 | Sem_070103_RelationalOperators_045 | NaN special float is evaulated correctly in a relation. | Clause 7.1.3 | m | y |
| 54 | Sem_070103_RelationalOperators_046 | NaN special float is evaulated correctly in a relation. | Clause 7.1.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 55 | Sem_070103_RelationalOperators_047 | Infinity special float is evaulated correctly in a relation. | Clause 7.1.3 | m | y |
| 56 | Sem_070103_RelationalOperators_048 | anytypes can be compared | Clause 7.1.3 | m | y |
| 57 | Sem_070103_RelationalOperators_049 | anytypes can be compared | Clause 7.1.3 | m | y |
| 58 | Sem_070103_RelationalOperators_050 | the less than or equal to operator on enumerations is evaluated correctly with differing values | Clause 7.1.3 | m | n |

# 5.52. Logical operators

*Table 51. Logical operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070104_LogicalOperators_001 | The boolean operator supports negation. | Clause 7.1.4 | m | y |
| 2 | Sem_070104_LogicalOperators_002 | The the and operator with true and false as operands work on boolean variables. | Clause 7.1.4 | m | y |

# 5.53. Bitwise operators

*Table 52. Bitwise operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070105_BitwiseOperators_001 | The bitwise negation operator works as expected. | Clause 7.1.5 | m | y |
| 2 | Sem_070105_BitwiseOperators_002 | The bitwise negation operator works as expected on hexstrings. | Clause 7.1.5 | m | y |

## 5.54. Shift operators

*Table 53. Shift operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070106_ShiftOperators_001 | The shift left operator works as expected on bitstrings. | Clause 7.1.6 | m | y |
| 2 | Sem_070106_ShiftOperators_002 | The shift left operator works as expected on hexstrings. | Clause 7.1.6 | m | y |
| 3 | Sem_070106_ShiftOperators_003 | The shift right operator works as expected on bitstrings. | Clause 7.1.6 | m | y |
| 4 | Sem_070106_ShiftOperators_004 | The shift right operator works as expected on hexstrings. | Clause 7.1.6 | m | y |

## 5.55. Rotate operators

*Table 54. Rotate operators*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_070107_RotateOperators_001 | The rotate left operator works as expected on bitstrings. | Clause 7.1.7 | m | y |
| 2 | Sem_070107_RotateOperators_002 | The rotate left operator works as expected on hexstrings. | Clause 7.1.7 | m | y |
| 3 | Sem_070107_RotateOperators_003 | The rotate right operator works as expected on bitstrings. | Clause 7.1.7 | m | y |
| 4 | Sem_070107_RotateOperators_004 | The rotate right operator works as expected on hexstrings. | Clause 7.1.7 | m | y |

## 5.56. Field references and list elements

*Table 55. Field references and list elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_0702_FieldReferencesAndListElements_001 | The IUT correctly handles field referencing | Clause 7.2 | m | y |
| 2 | Sem_0702_FieldReferencesAndListElements_002 | The IUT correctly handles field referencing | Clause 7.2 | m | y |

# 5.57. Definition of a module

*Table 56. Definition of a module*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_0801_DefinitionOfAModule_001 | A module definition with multiple language specifications is rejected. | Clause 8.1 | m | n |
| 2 | Syn_0801_DefinitionOfAModule_001 | A ``plain'' module definition is accepted. | Clause 8.1 | m | y |
| 3 | Syn_0801_DefinitionOfAModule_002 | A module definition with language specification is accepted. | Clause 8.1 | m | y |
| 4 | Syn_0801_DefinitionOfAModule_003 | A module definition with language and package is accepted. | Clause 8.1 | m | n |
| 5 | Syn_0801_DefinitionOfAModule_004 | A module definition with package and without language is accepted. | Clause 8.1 | m | y |
| 6 | Syn_0801_DefinitionOfAModule_005 | A module definition with ed4.3.1 language and package is accepted. | Clause 8.1 | m | y |
| 7 | Syn_0801_DefinitionOfAModule_006 | A module definition with ed4.4.1 language and package is accepted. | Clause 8.1 | m | y |
| 8 | Syn_0801_DefinitionOfAModule_007 | A module definition with ed4.5.1 language and package is accepted. | Clause 8.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | Syn_0801_DefinitionOfAModule_008 | A module definition with ed4.6.1 language and package is accepted. | Clause 8.1 | m | y |
| 10 | Syn_0801_DefinitionOfAModule_009 | A module definition with ed4.7.1 language and package is accepted. | Clause 8.1 | m | y |
| 11 | Syn_0801_DefinitionOfAModule_010 | A module definition with ed4.8.1 language and package is accepted | Clause 8.1 | m | y |

# 5.58. Module definitions part

*Table 57. Module definitions part*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_0802_ModuleDefinitionsPart_001 | A TypeDef module definition with public visibility is accepted. | Clause 8.2 | m | y |
| 2 | Syn_0802_ModuleDefinitionsPart_002 | A TypeDef module definition with private visibility is accepted. | Clause 8.2 | m | y |

# 5.59. Module parameters

*Table 58. Module parameters*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_080201_ModuleParameters_001 | A port, default or component types cannot be module parameters | Clause 8.2.1 | m | y |
| 2 | NegSem_080201_ModuleParameters_002 | A port, default or component types cannot be module parameters | Clause 8.2.1 | m | n |
| 3 | NegSem_080201_ModuleParameters_003 | A port, default or component types cannot be module parameters | Clause 8.2.1 | m | n |
| 4 | NegSem_080201_ModuleParameters_004 | Ensure that module parameters remain constant | Clause 8.2.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSem_080201_ModuleParameters_005 | A reference to plain module parameter with a default value delivers the default value unless it is overwritten | Clause 8.2.1 | m | y |
| 6 | NegSem_080201_ModuleParameters_006 | A reference to plain module parameter with a default value delivers the default value unless it is overwritten | Clause 8.2.1 | m | y |
| 7 | NegSyn_080201_ModuleParameters_001 | Module parameter can be declared within the module definition part only | Clause 8.2.1 | m | y |
| 8 | NegSyn_080201_ModuleParameters_002 | Module parameter can be declared within the module definition part only | Clause 8.2.1 | m | y |
| 9 | Sem_080201_ModuleParameters_001 | A reference to plain module parameter with a default value delivers the default value unless it is overwritten. | Clause 8.2.1 | m | y |
| 10 | Syn_080201_ModuleParameters_001 | Plain module parameters are accepted. | Clause 8.2.1 | m | y |
| 11 | Syn_080201_ModuleParameters_002 | Plain module parameters with default values are accepted. | Clause 8.2.1 | m | y |
| 12 | Syn_080201_ModuleParameters_003 | Plain module parameters with default values and visibility modifiers are accepted. | Clause 8.2.1 | m | y |

# 5.60. Groups of definitions

*Table 59. Groups of definitions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_080202_GroupOfDefinitions_001 | A definition within a group is accepted. | Clause 8.2.2 | m | y |
| 2 | Syn_080202_GroupOfDefinitions_002 | A definition within a nested group is accepted. | Clause 8.2.2 | m | y |
| 3 | Syn_080202_GroupOfDefinitions_003 | A definition within a group with public visibility modifier is accepted. | Clause 8.2.2 | m | y |
| 4 | Syn_080202_GroupOfDefinitions_004 | A definition within a group with public visibility modifier and attributes is accepted. | Clause 8.2.2 | m | y |

# 5.61. General format of import

*Table 60. General format of import*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020301_GeneralFormatOfImport_001 | Name handling of imported enumerations is properly handled | Clause 8.2.3.1 | m | n |
| 2 | NegSem_08020301_GeneralFormatOfImport_002 | Name handling of imported enumerations is properly handled | Clause 8.2.3.1 | m | y |
| 3 | NegSem_08020301_GeneralFormatOfImport_005 | Make sure that the identifier of the current module cannot be used for prefixing imported entities | Clause 8.2.3.1 | m | y |
| 4 | NegSem_08020301_GeneralFormatOfImport_006 | The only top-level visible definitions of a module may be imported. | Clause 8.2.3.1 | m | y |
| 5 | NegSem_08020301_GeneralFormatOfImport_007 | Verify that information about message types is imported together with port type | Clause 8.2.3.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_08020301_GeneralFormatOfImport_008 | Verify that identifiers of module parameter types are not imported together with module parameters | Clause 8.2.3.1 | m | n |
| 7 | NegSem_08020301_GeneralFormatOfImport_009 | Verify that identifiers of constant types are not imported together with constants | Clause 8.2.3.1 | m | n |
| 8 | NegSem_08020301_GeneralFormatOfImport_010 | Verify that identifiers of field types are not imported together with structured types | Clause 8.2.3.1 | m | n |
| 9 | NegSem_08020301_GeneralFormatOfImport_011 | Verify that identifiers of message types are not imported together with port types | Clause 8.2.3.1 | m | n |
| 10 | NegSem_08020301_GeneralFormatOfImport_012 | Verify that identifiers of signatures are not imported together with port types | Clause 8.2.3.1 | m | n |
| 11 | NegSem_08020301_GeneralFormatOfImport_013 | Verify that identifiers of constant types are not imported together with component types | Clause 8.2.3.1 | m | n |
| 12 | NegSem_08020301_GeneralFormatOfImport_014 | Verify that identifiers of variable types are not imported together with component types | Clause 8.2.3.1 | m | n |
| 13 | NegSem_08020301_GeneralFormatOfImport_015 | Verify that identifiers of port types are not imported together with component types | Clause 8.2.3.1 | m | n |
| 14 | NegSem_08020301_GeneralFormatOfImport_016 | Verify that identifiers of parameter types are not imported together with signatures | Clause 8.2.3.1 | m | n |
| 15 | NegSem_08020301_GeneralFormatOfImport_017 | Verify that identifiers of return types are not imported together with signatures | Clause 8.2.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 16 | NegSem_08020301_GeneralFormatOfImport_018 | Verify that identifiers of exception types are not imported together with signatures | Clause 8.2.3.1 | m | n |
| 17 | NegSem_08020301_GeneralFormatOfImport_019 | Verify that identifiers of template types are not imported together with data templates | Clause 8.2.3.1 | m | n |
| 18 | NegSem_08020301_GeneralFormatOfImport_020 | Verify that identifiers of parameter types are not imported together with data templates | Clause 8.2.3.1 | m | n |
| 19 | NegSem_08020301_GeneralFormatOfImport_021 | Verify that identifiers of constants are not imported together with data templates | Clause 8.2.3.1 | m | n |
| 20 | NegSem_08020301_GeneralFormatOfImport_022 | Verify that identifiers of module parameters are not imported together with data templates | Clause 8.2.3.1 | m | n |
| 21 | NegSem_08020301_GeneralFormatOfImport_023 | Verify that identifiers of functions are not imported together with data templates | Clause 8.2.3.1 | m | n |
| 22 | NegSem_08020301_GeneralFormatOfImport_024 | Verify that identifiers of signatures are not imported together with signature templates | Clause 8.2.3.1 | m | n |
| 23 | NegSem_08020301_GeneralFormatOfImport_025 | Verify that identifiers of constants are not imported together with signature templates | Clause 8.2.3.1 | m | n |
| 24 | NegSem_08020301_GeneralFormatOfImport_026 | Verify that identifiers of module parameters are not imported together with signature templates | Clause 8.2.3.1 | m | n |
| 25 | NegSem_08020301_GeneralFormatOfImport_027 | Verify that identifiers of functions are not imported together with signature templates | Clause 8.2.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 26 | NegSem_08020301_GeneralFormatOfImport_028 | Verify that identifiers of parameter types are not imported together with functions | Clause 8.2.3.1 | m | n |
| 27 | NegSem_08020301_GeneralFormatOfImport_029 | Verify that identifiers of return type are not imported together with functions | Clause 8.2.3.1 | m | n |
| 28 | NegSem_08020301_GeneralFormatOfImport_030 | Verify that identifiers of component types are not imported together with functions | Clause 8.2.3.1 | m | n |
| 29 | NegSem_08020301_GeneralFormatOfImport_031 | Verify that identifiers of parameter types are not imported together with external functions | Clause 8.2.3.1 | m | n |
| 30 | NegSem_08020301_GeneralFormatOfImport_032 | Verify that identifiers of return type are not imported together with external functions | Clause 8.2.3.1 | m | n |
| 31 | NegSem_08020301_GeneralFormatOfImport_033 | Verify that identifiers of parameter types are not imported together with altsteps | Clause 8.2.3.1 | m | n |
| 32 | NegSem_08020301_GeneralFormatOfImport_034 | Verify that identifiers of component types are not imported together with altsteps | Clause 8.2.3.1 | m | n |
| 33 | NegSem_08020301_GeneralFormatOfImport_035 | Verify that identifiers of parameter types are not imported together with test cases | Clause 8.2.3.1 | m | n |
| 34 | NegSem_08020301_GeneralFormatOfImport_036 | Verify that identifiers of component types (`runs on`) are not imported together with test cases | Clause 8.2.3.1 | m | n |
| 35 | NegSem_08020301_GeneralFormatOfImport_037 | Verify that identifiers of component types (`system`) are not imported together with test cases | Clause 8.2.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 36 | NegSem_08020301_GeneralFormatOfImport_038 | Verify that definition from inside an imported function cannot be referenced | Clause 8.2.3.1 | m | y |
| 37 | NegSem_08020301_GeneralFormatOfImport_039 | Verify that `import` clause cannot override language tag of imported module | Clause 8.2.3.1 | m | n |
| 38 | NegSem_08020301_GeneralFormatOfImport_040 | Verify that unsupported language concepts cannot be used when language is set by `import` clause | Clause 8.2.3.1 | m | n |
| 39 | NegSyn_08020301_GeneralFormatOfImport_001 | `import` statement cannot be used in test case blocks | Clause 8.2.3.1 | m | y |
| 40 | NegSyn_08020301_GeneralFormatOfImport_002 | `import` statement cannot be used in module control part | Clause 8.2.3.1 | m | y |
| 41 | Sem_08020301_GeneralFormatOfImport_003 | Make sure that local definition takes precedence over imported one when their identifiers are equal | Clause 8.2.3.1 | m | y |
| 42 | Sem_08020301_GeneralFormatOfImport_004 | Make sure that imported enumeration values take precedence over local definition | Clause 8.2.3.1 | m | y |
| 43 | Sem_08020301_GeneralFormatOfImport_005 | Make sure that it is possible to use module prefix for local definitions | Clause 8.2.3.1 | m | y |
| 44 | Sem_08020301_GeneralFormatOfImport_006 | Make sure that it is possible to use module prefix for local definitions | Clause 8.2.3.1 | m | n |
| 45 | Sem_08020301_GeneralFormatOfImport_007 | Make sure that it is possible to use module prefix for imported definitions | Clause 8.2.3.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 46 | Sem_08020301_GeneralFormatOfImport_008 | Verify that structured type is imported together with its field names and nested type definitions | Clause 8.2.3.1 | m | y |
| 47 | Sem_08020301_GeneralFormatOfImport_009 | Verify that component type is imported together with constant, variable, timer and port names | Clause 8.2.3.1 | m | y |
| 48 | Sem_08020301_GeneralFormatOfImport_010 | Verify that signature is imported together with parameter names | Clause 8.2.3.1 | m | y |
| 49 | Sem_08020301_GeneralFormatOfImport_011 | Verify that parameterized template is imported together with parameter names | Clause 8.2.3.1 | m | y |
| 50 | Sem_08020301_GeneralFormatOfImport_012 | Verify that function is imported together with parameter names | Clause 8.2.3.1 | m | y |
| 51 | Sem_08020301_GeneralFormatOfImport_013 | Verify that altstep is imported together with parameter names | Clause 8.2.3.1 | m | y |
| 52 | Sem_08020301_GeneralFormatOfImport_014 | Verify that test case is imported together with parameter names | Clause 8.2.3.1 | m | y |
| 53 | Sem_08020301_GeneralFormatOfImport_015 | Verify that information about module parameter type is imported together with module parameter | Clause 8.2.3.1 | m | y |
| 54 | Sem_08020301_GeneralFormatOfImport_016 | Verify that information about type of constant is imported together with constant | Clause 8.2.3.1 | m | y |
| 55 | Sem_08020301_GeneralFormatOfImport_017 | Verify using of `import` clause with language tag for importing module having identical language tag | Clause 8.2.3.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 56 | Sem_08020301_GeneralFormatOfImport_018 | Verify using of `import` clause with language tag for importing module with no language tag | Clause 8.2.3.1 | m | y |
| 57 | Sem_08020301_GeneralFormatOfImport_019 | Verify that type of port is imported from a module as expected | Clause 8.2.3.1 | m | y |
| 58 | Sem_08020301_GeneralFormatOfImport_020 | Verify that prefixed type is evaluated as expected | Clause 8.2.3.1 | m | y |
| 59 | Syn_08020301_GeneralFormatOfImport_001 | Import all is accepted. | Clause 8.2.3.1 | m | y |
| 60 | Syn_08020301_GeneralFormatOfImport_002 | Import of specific types is accepted. | Clause 8.2.3.1 | m | n |

# 5.62. Importing single definitions

*Table 61. Importing single definitions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_08020302_ImportingSingleDefinitions_001 | The value of an explicitly imported constant can be read and carries the same value. | Clause 8.2.3.2 | m | n |
| 2 | Sem_08020302_ImportingSingleDefinitions_002 | The value of an explicitly imported template can be read and carries the same value. | Clause 8.2.3.2 | m | n |

# 5.63. Importing groups

*Table 62. Importing groups*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020303_ImportingGroups_001 | Constants listed as exceptions in imported groups are not accessible. | Clause 8.2.3.3 | m | n |
| 2 | Sem_08020303_ImportingGroups_001 | A const defined in a group can be accessed if the group is imported. | Clause 8.2.3.3 | m | n |
| 3 | Sem_08020303_ImportingGroups_002 | The IUT properly handles `except` clause in group import definitions | Clause 8.2.3.3 | m | n |
| 4 | Sem_08020303_ImportingGroups_003 | but that it is in fact a shortcut notation for explicit imports. | Clause 8.2.3.3 | m | n |

# 5.64. Importing definitions of the same kind

*Table 63. Importing definitions of the same kind*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020301_GeneralFormatOfImport_003 | Transitive import rules are properly handled | Clause 8.2.3.4 | m | y |
| 2 | NegSem_08020301_GeneralFormatOfImport_004 | Transitive import rules are properly handled | Clause 8.2.3.4 | m | y |
| 3 | Sem_08020301_GeneralFormatOfImport_001 | Transitive imports are properly handled | Clause 8.2.3.4 | m | y |
| 4 | Sem_08020301_GeneralFormatOfImport_002 | Enumerated type definitions are automatically imported when needed | Clause 8.2.3.4 | m | y |
| 5 | Sem_08020304_ImportingDefinitionsOfTheSameKind_001 | An import of all constants allows access to a sample constant. | Clause 8.2.3.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_08020304_ImportingDefinitionsOfTheSameKind_002 | A previously valid const import is not removed by an import covering the same definition with an except. | Clause 8.2.3.4 | m | n |
| 7 | Sem_08020304_ImportingDefinitionsOfTheSameKind_003 | A previously valid const import is not removed by a second `import` statement excluding the same definition. | Clause 8.2.3.4 | m | n |

# 5.65. Importing all definitions of a module

*Table 64. Importing all definitions of a module*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020305_ImportingAllDefinitionsOfAModule_001 | The constant is not visible after import with except. | Clause 8.2.3.5 | m | n |
| 2 | NegSem_08020305_ImportingAllDefinitionsOfAModule_002 | The constant is not visible after import with except. | Clause 8.2.3.5 | m | n |
| 3 | Sem_08020305_ImportingAllDefinitionsOfAModule_001 | The constant is be visible after multiple imports. | Clause 8.2.3.5 | m | y |
| 4 | Sem_08020305_ImportingAllDefinitionsOfAModule_002 | The constant is be visible after multiple imports. | Clause 8.2.3.5 | m | n |

# 5.66. Import definitions from other TTCN-3 editions and from non-TTCN-3 modules

*Table 65. Import definitions from other TTCN-3 editions and from non-TTCN-3 modules*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_08020306_ImportingDefinitionsFromOtherT3EditionsAndFromNonT3Modules_001 | It is possible to import from previous language versions. | Clause 8.2.3.6 | m | y |
| 2 | Syn_08020306_ImportingDefinitionsFromOtherT3EditionsAndFromNonT3Modules_001 | Imports work with language references when importing definitions of the same kinds (in this case constants) is accepted. | Clause 8.2.3.6 | m | y |
| 3 | Syn_08020306_ImportingDefinitionsFromOtherT3EditionsAndFromNonT3Modules_002 | Imports work with language references when importing all definitions of another module is accepted. | Clause 8.2.3.6 | m | y |

## 5.67. Importing of `import` statements from TTCN-3 modules

*Table 66. Importing of `import` statements from TTCN-3 modules*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020307_ImportingOfImportStatementsFromT3Modules_001 | The import of `import` statements works for import all. | Clause 8.2.3.7 | m | y |
| 2 | NegSem_08020307_ImportingOfImportStatementsFromT3Modules_002 | The import of `import` statements works for import all. | Clause 8.2.3.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | Sem_08020307_ImportingOfImportStatementsFromT3Modules_001 | The import of `import` statements works for import all. | Clause 8.2.3.7 | m | y |

# 5.68. Compatibility of language specifications of imports

*Table 67. Compatibility of language specifications of imports*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_08020308_CompatibilityOfLanguageSpecificationsInImports_001 | Imports referring to future TTCN-3 versions are rejected. | Clause 8.2.3.8 | m | n |
| 2 | NegSem_08020308_CompatibilityOfLanguageSpecificationsInImports_002 | Verify that modules with explicit language tag cannot import from newer TTCN-3 versions | Clause 8.2.3.8 | m | n |
| 3 | NegSem_08020308_CompatibilityOfLanguageSpecificationsInImports_003 | Verify that modules with explicit language tag cannot import from newer TTCN-3 versions | Clause 8.2.3.8 | m | n |
| 4 | Sem_08020308_CompatibilityOfLanguageSpecificationsInImports_001 | Verify that modules with explicit language tag can import from older TTCN-3 versions | Clause 8.2.3.8 | m | y |
| 5 | Sem_08020308_CompatibilityOfLanguageSpecificationsInImports_002 | Verify that modules with explicit language tag can import from older TTCN-3 versions | Clause 8.2.3.8 | m | y |

# 5.69. Definition of friend modules

*Table 68. Definition of friend modules*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_080204_DefinitionOfFriendModules_001 | Friend visibility works for a sample constant. | Clause 8.2.4 | m | y |
| 2 | NegSem_080204_DefinitionOfFriendModules_002 | Private definitions are not made visible by friend declarations (for a constant sample definition). | Clause 8.2.4 | m | y |
| 3 | Sem_080204_DefinitionOfFriendModules_001 | Friend visibility works for a sample constant. | Clause 8.2.4 | m | y |

# 5.70. Visibility of definitions

*Table 69. Visibility of definitions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_080205_VisibilityOfDefinitions_001 | Private definition (in this case a sample constant) is not visible using a normal import. | Clause 8.2.5 | m | y |
| 2 | NegSem_080205_VisibilityOfDefinitions_002 | Private definition (in this case a sample constant) is not visible using an import of a friend module. | Clause 8.2.5 | m | y |
| 3 | NegSem_080205_VisibilityOfDefinitions_003 | Friend definition (in this case a sample constant) is not visible using a group import of a non-friend module. | Clause 8.2.5 | m | y |
| 4 | NegSem_080205_VisibilityOfDefinitions_004 | Private definition (in this case a sample constant) is not visible using a group import of a non-friend module. | Clause 8.2.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSem_080205_VisibilityOfDefinitions_005 | Private definition (in this case a sample constant) is not visible using a group import of a friend module. | Clause 8.2.5 | m | y |
| 6 | Sem_080205_VisibilityOfDefinitions_001 | Explicitly defined public definitions (in this case a sample constant) are visible when imported. | Clause 8.2.5 | m | y |
| 7 | Sem_080205_VisibilityOfDefinitions_002 | Explicitly defined public definitions (in this case a sample constant) are visible when imported by a friend module. | Clause 8.2.5 | m | y |
| 8 | Sem_080205_VisibilityOfDefinitions_003 | Explicitly defined public definitions (in this case a sample constant) are visible when imported through a group. | Clause 8.2.5 | m | y |
| 9 | Sem_080205_VisibilityOfDefinitions_004 | Explicitly defined public definitions (in this case a sample constant) are visible when imported through a group of a friend module. | Clause 8.2.5 | m | y |
| 10 | Sem_080205_VisibilityOfDefinitions_005 | Friend definitions (in this case a sample constant) are visible when imported through a group of a friend module. | Clause 8.2.5 | m | y |

# 5.71. Module control part

*Table 70. Module control part*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_0803_ModuleControlPart_001 | There is not more than one control part. | Clause 8.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_0803_ModuleControlPart_001 | The verdict returned from a test case to the control-part does not influence the execution of a second test case. The result of the last test case execution corresponds to the overall test verdict. | Clause 8.3 | m | y |
| 3 | Syn_0803_ModuleControlPart_001 | The module control is able to accept `execute` statements. | Clause 8.3 | m | y |
| 4 | Syn_0803_ModuleControlPart_002 | The module control part with a few commonly used stateents is accepted. | Clause 8.3 | m | y |
| 5 | Syn_0803_ModuleControlPart_003 | An empty control part is accepted. | Clause 8.3 | m | y |

# 5.72. Port types, component types and test configurations

*Table 71. Port types, component types and test configurations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_0901_Communication_ports_001 | The IUT correctly handles loopback message | Clause 9 | m | y |
| 2 | Sem_0901_Communication_ports_002 | The the IUT receives the message sent by mycompA | Clause 9 | m | y |
| 3 | Sem_0901_Communication_ports_003 | The the IUT receives the message sent by mycompB and mycompC | Clause 9 | m | y |
| 4 | Sem_0901_Communication_ports_004 | The IUT correctly handles message exch. between ports | Clause 9 | m | y |
| 5 | Sem_0901_Communication_ports_005 | The the IUT receives the message sent by mycompA | Clause 9 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_0902_Communication_ports_001 | The IUT correctly handles the assoc. of two port to the same system interface | Clause 9 | m | n |
| 7 | NegSem_0902_Communication_ports_002 | The mycomp is connected to two system interface port. | Clause 9 | m | n |
| 8 | NegSem_0902_Communication_ports_003 | The two system interf. port cannot connect | Clause 9 | m | y |
| 9 | NegSem_0902_Communication_ports_004 | The a connected port cannot be mapped | Clause 9 | m | n |
| 10 | Sem_0902_Communication_ports_001 | The IUT port correctly mapped with a system interface | Clause 9 | m | y |
| 11 | Sem_0902_Communication_ports_002 | The IUTs two ports are mapped correctly to system interfaces | Clause 9 | m | y |
| 12 | Syn_0902_Communication_ports_001 | Two component can be mapped by one system interface | Clause 9 | m | y |

# 5.73. Communication ports

*Table 72. Communication ports*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_0901_Communication_ports_001 | A port owned by a component cannot be connected with two other ports | Clause 9.1 | m | n |
| 2 | NegSem_0901_Communication_ports_002 | It is not possible to connect a mapped port | Clause 9.1 | m | y |
| 3 | NegSem_0901_Communication_ports_003 | It is not possible to connect a port with two other ports owned by the same component | Clause 9.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_0901_Communication_ports_004 | Verify that it is not possible to map a connected port | Clause 9.1 | m | y |
| 5 | NegSem_0901_Communication_ports_005 | Verify that it is not possible to connect a port with a port owned by the same component | Clause 9.1 | m | n |
| 6 | NegSem_0901_Communication_ports_006 | Verify that only 1:1 connection between component port and TSI are allowed | Clause 9.1 | m | n |
| 7 | NegSem_0901_Communication_ports_007 | Verify that a two TSI port cannot be connected | Clause 9.1 | m | y |
| 8 | NegSem_0901_Communication_ports_008 | Verify that mapping an already connected port is not allowed | Clause 9.1 | m | n |
| 9 | NegSem_0901_Communication_ports_009 | Verify that connections within the test system interface are not allowed | Clause 9.1 | m | y |
| 10 | NegSyn_0901_Communication_ports_001 | Verify that a two TSI port cannot be connected | Clause 9.1 | m | y |
| 11 | Sem_0901_Communication_ports_006 | Verify that a port can connect to itself | Clause 9.1 | m | y |
| 12 | Sem_0901_Communication_ports_007 | Verify that a port can connect to another port of the same component | Clause 9.1 | m | y |
| 13 | Sem_0901_Communication_ports_008 | Verify that more than one component port can mapped to a single system port | Clause 9.1 | m | y |
| 14 | Sem_0901_Communication_ports_009 | Verify that a component port can be connected to two other component ports | Clause 9.1 | m | y |
| 15 | Sem_0901_Communication_ports_010 | Verify that a component port can be mapped to TSI port | Clause 9.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 16 | Sem_0901_Communication_ports_011 | Verify that a component ports can be mapped to TSI ports | Clause 9.1 | m | y |

# 5.74. Declaring constants

*Table 73. Declaring constants*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_10_Constants_001 | Assign rnd to constant used in type, not allowed since constant expressions used in types have to be known at compile-time. | Clause 10 | m | y |
| 2 | NegSem_10_Constants_002 | A value is assigned only once to a constant | Clause 10 | m | y |
| 3 | NegSem_10_Constants_003 | Constant shall not be of port type | Clause 10 | m | y |
| 4 | NegSem_10_Constants_004 | Dot notation of a field in a record, which actual value is `null` shall cause an error | Clause 10 | m | n |
| 5 | NegSem_10_Constants_005 | Index notation of a field in a set of type, which actual value is `null` shall cause an error | Clause 10 | m | n |
| 6 | Sem_10_Constants_001 | Assign and read constants | Clause 10 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Sem_10_Constants_002 | Assign and read constants values | Clause 10 | m | y |
| 8 | Sem_10_Constants_003 | Single expression and constant values | Clause 10 | m | y |
| 9 | Sem_10_Constants_004 | Constant used within invoke function with return | Clause 10 | m | y |
| 10 | Sem_10_Constants_005 | Constant used within predefined function | Clause 10 | m | y |
| 11 | Sem_10_Constants_006 | Record type used as a constant | Clause 10 | m | y |
| 12 | Sem_10_Constants_007 | Record type used as a constant with optional fields | Clause 10 | m | y |
| 13 | Sem_10_Constants_008 | Set type used as a constant | Clause 10 | m | y |
| 14 | Sem_10_Constants_009 | Set type used as a constant with optional fields | Clause 10 | m | y |
| 15 | Syn_10_Constants_001 | Create constants | Clause 10 | m | y |
| 16 | Syn_10_Constants_002 | Assign default constants values | Clause 10 | m | y |
| 17 | Syn_10_Constants_003 | Assign component constants values | Clause 10 | m | y |
| 18 | Syn_10_Constants_004 | Define constants in different scopes | Clause 10 | m | y |

# 5.75. Value variables

*Table 74. Value variables*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1101_ValueVars_001 | Variables should be assigned only by values | Clause 11.1 | m | y |
| 2 | NegSem_1101_ValueVars_002 | Partially initialized variables are evaluated correctly. | Clause 11.1 | m | n |
| 3 | NegSem_1101_ValueVars_003 | Dot notation referencing to a field, which actual value is `null` shall cause an error. | Clause 11.1 | m | n |
| 4 | NegSem_1101_ValueVars_004 | Index notation referencing to a "set of", which actual value is `null` shall cause an error. | Clause 11.1 | m | n |
| 5 | NegSem_1101_ValueVars_005 | Variables should be assigned only by values | Clause 11.1 | m | y |
| 6 | NegSyn_1101_ValueVars_001 | Define variables in module scope | Clause 11.1 | m | y |
| 7 | Sem_1101_ValueVars_001 | Define variables in different scopes | Clause 11.1 | m | y |
| 8 | Sem_1101_ValueVars_002 | Define variables in different scopes | Clause 11.1 | m | y |
| 9 | Sem_1101_ValueVars_003 | Read and write variables | Clause 11.1 | m | y |
| 10 | Sem_1101_ValueVars_004 | Partially initialized variables are evaluated correctly. | Clause 11.1 | m | y |
| 11 | Sem_1101_ValueVars_005 | Partially initialized variables are evaluated correctly. | Clause 11.1 | m | y |
| 12 | Syn_1101_ValueVars_001 | Define variables in different scopes | Clause 11.1 | m | y |

# 5.76. Template variables

*Table 75. Template variables*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1102_TemplateVars_001 | Template variables should be assigned with unitialized variables | Clause 11.2 | m | y |
| 2 | NegSem_1102_TemplateVars_002 | Partially initialized templates are evaluated correctly. | Clause 11.2 | m | n |
| 3 | NegSem_1102_TemplateVars_003 | Dot notation referencing to a field, which actual value is `null` shall cause an error. | Clause 11.2 | m | n |
| 4 | NegSem_1102_TemplateVars_004 | Index notation referencing to a set of, which actual value is `null` shall cause an error. | Clause 11.2 | m | n |
| 5 | NegSyn_1102_TemplateVars_001 | Define template variables in module scope | Clause 11.2 | m | y |
| 6 | NegSyn_1102_TemplateVars_002 | Template variables should be assigned with unitialized variables | Clause 11.2 | m | y |
| 7 | Sem_1102_TemplateVars_001 | Define variables in different scopes | Clause 11.2 | m | y |
| 8 | Sem_1102_TemplateVars_002 | Partially initialized templates are evaluated correctly. | Clause 11.2 | m | y |
| 9 | Sem_1102_TemplateVars_003 | Partially initialized templates are evaluated correctly. | Clause 11.2 | m | y |
| 10 | Syn_1102_TemplateVars_001 | Define template variables in different scopes | Clause 11.2 | m | y |

## 5.77. Declaring timers

*Table 76. Declaring timers*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_12_toplevel_timer_001 | Ensure timer can not be initialized with negative duration | Clause 12 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_12_toplevel_timer_002 | Ensure timer in array can not be initialized with negative duration | Clause 12 | m | y |
| 3 | NegSem_12_toplevel_timer_003 | Ensure uninitialized timer can’t be started | Clause 12 | m | y |
| 4 | NegSem_12_toplevel_timer_004 | Ensure uninitialized timer in array can’t be started | Clause 12 | m | y |
| 5 | NegSem_12_toplevel_timer_005 | Ensure uninitialized timer in array can’t be started | Clause 12 | m | y |
| 6 | NegSem_12_toplevel_timer_006 | Ensure `timer declaration syntax` - reject single timer instance initialized with array | Clause 12 | m | y |
| 7 | NegSem_12_toplevel_timer_007 | Ensure `timer declaration` syntax – reject array initialization with wrong number of initializers | Clause 12 | m | y |
| 8 | NegSem_12_toplevel_timer_008 | Ensure `timer declaration` syntax – reject array of timers initizlized with a single float value | Clause 12 | m | y |
| 9 | NegSyn_12_toplevel_timer_001 | Ensure timer can`t be used in module control parts when declared in components | Clause 12 | m | y |
|  | 10 | NegSyn_12_toplevel_timer_002 | Ensure `timer declaration` syntax | Clause 12 | m |
| y |  | 11 | NegSyn_12_toplevel_timer_003 | Ensure `timer declaration` syntax | Clause 12 |
| m | y |  | 12 | NegSyn_12_toplevel_timer_004 | Ensure `timer declaration` syntax |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| Clause 12 | m | y | | 13 | NegSyn_12_toplevel_timer_005 |
| Ensure `timer declaration` syntax | Clause 12 | m | y | | 14 |
| NegSyn_12_toplevel_timer_006 | Ensure timer array declaration syntax | Clause 12 | m | y | |
| 15 | NegSyn_12_toplevel_timer_007 | Ensure `timer array declaration` syntax | Clause 12 | m | y |
| | 16 | Sem_12_toplevel_timer_001 | Ensure timer can be declared in components | Clause 12 | m |
| y | | 17 | Sem_12_toplevel_timer_002 | Ensure timer can be declared in module control parts | Clause 12 |
| m | y | | 18 | Sem_12_toplevel_timer_003 | Ensure timer can be declared in altsteps |
| Clause 12 | m | y | | 19 | Sem_12_toplevel_timer_004 |
| Ensure timer can be declared in functions | Clause 12 | m | y | | 20 |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| Sem_12_toplevel_timer_005 | Ensure timer can be declared in test cases | Clause 12 | m | y | |
| 21 | Sem_12_toplevel_timer_006 | Ensure timer`s elapsed time is plausible | Clause 12 | m | y |
| 22 | Sem_12_toplevel_timer_007 | Ensure timer can be declared in components but used in test cases | Clause 12 | m | y |
| 23 | Sem_12_toplevel_timer_008 | Ensure timer can be declared in components but used in functions | Clause 12 | m | y |
| 24 | Sem_12_toplevel_timer_009 | Ensure timer can be declared in components but used in altsteps | Clause 12 | m | y |
| 25 | Syn_12_toplevel_timer_001 | Ensure non-initialized `timer declaration` syntax | Clause 12 | m | y |
| 26 | Syn_12_toplevel_timer_002 | Ensure `timer array declaration` syntax | Clause 12 | m | y |
| 27 | Syn_12_toplevel_timer_003 | Ensure definition of a list of timers is allowed as a single declaration | Clause 12 | m | y |
| 28 | Syn_12_toplevel_timer_004 | Ensure `timer array initialization` syntax | Clause 12 | m | y |
| 29 | Syn_12_toplevel_timer_005 | Ensure timer declaration with expression | Clause 12 | m | y |
| 30 | Syn_12_toplevel_timer_006 | Ensure timer declaration with expression | Clause 12 | m | y |

# 5.78. Declaring messages

*Table 77. Declaring messages*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_13_declaring_msg_001 | Ensure received messages can be a combination of value and matching mechanism | Clause 13 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_13_declaring_msg_002 | Ensure received messages can`t be matched with wrong template | Clause 13 | m | y |
| 3 | Sem_13_declaring_msg_003 | Ensure instances of messages can be declared by in-line templates | Clause 13 | m | y |
| 4 | Sem_13_declaring_msg_004 | Ensure instances of messages can be declared by global templates | Clause 13 | m | y |
| 5 | Sem_13_declaring_msg_005 | Ensure instances of messages can be declared and passed via template variables | Clause 13 | m | y |
| 6 | Sem_13_declaring_msg_006 | Ensure instances of messages can be declared and passed via inline template | Clause 13 | m | y |
| 7 | Sem_13_declaring_msg_007 | Ensure instances of messages can be declared and passed via parameter | Clause 13 | m | y |
| 8 | Sem_13_declaring_msg_008 | Ensure instances of messages can be declared and passed via template parameter | Clause 13 | m | y |
| 9 | Sem_13_declaring_msg_009 | Ensure instances of messages can be declared and passed via template parameter | Clause 13 | m | y |
| 10 | Sem_13_toplevel_declaring_msg_various_types_001 | Port with type anytype can send and receive messages of any basic or structured type: `record' type. | Clause 13 | m | y |
| 11 | Sem_13_toplevel_declaring_msg_various_types_002 | Port with type anytype can send and receive messages of any basic or structured type: `record of' type. | Clause 13 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 12 | Sem_13_toplevel_declaring_msg_various_types_003 | Port with type anytype can send and receive messages of any basic or structured type: `enum' type. | Clause 13 | m | y |
| 13 | Sem_13_toplevel_declaring_msg_various_types_004 | Port with type anytype can send and receive messages of any basic or structured type: `set' type. | Clause 13 | m | y |
| 14 | Sem_13_toplevel_declaring_msg_various_types_005 | Port with type anytype can send and receive messages of any basic or structured type: `union' type. | Clause 13 | m | y |
| 15 | Sem_13_toplevel_declaring_msg_various_types_006 | Port with type anytype can send and receive messages of any basic or structured type: `bitstring' type. | Clause 13 | m | y |
| 16 | Sem_13_toplevel_declaring_msg_various_types_007 | Port with type anytype can send and receive messages of any basic or structured type: `boolean' type. | Clause 13 | m | y |
| 17 | Sem_13_toplevel_declaring_msg_various_types_008 | Port with type anytype can send and receive messages of any basic or structured type: `charstring' type. | Clause 13 | m | y |
| 18 | Sem_13_toplevel_declaring_msg_various_types_009 | Port with type anytype can send and receive messages of any basic or structured type: `float' type. | Clause 13 | m | y |
| 19 | Sem_13_toplevel_declaring_msg_various_types_010 | Port with type anytype can send and receive messages of any basic or structured type: `hexstring' type. | Clause 13 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 20 | Sem_13_toplevel_declaring_msg_various_types_011 | Port with type anytype can send and receive messages of any basic or structured type: `integer' type. | Clause 13 | m | y |
| 21 | Sem_13_toplevel_declaring_msg_various_types_012 | Port with type anytype can send and receive messages of any basic or structured type: `octetstring' type. | Clause 13 | m | y |
| 22 | Sem_13_toplevel_declaring_msg_various_types_013 | Port with type anytype can send and receive messages of any basic or structured type: `universal charstring' type. | Clause 13 | m | n |
| 23 | Sem_13_toplevel_declaring_msg_various_types_014 | Port with type anytype can send and receive messages of any basic or structured type: `verdicttype' type. | Clause 13 | m | y |

# 5.79. Declaring procedure signatures

*Table 78. Declaring procedure signatures*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1400_procedure_signatures_001 | Nonblocking signature contains `in` parameter | Clause 14 | m | y |
| 1 | NegSem_1400_procedure_signatures_002 | Blocking calls needs response or exception handling | Clause 14 | m | y |
| 2 | Sem_1400_procedure_signatures_001 | The IUT calls signature exception | Clause 14 | m | y |
| 3 | Sem_1400_procedure_signatures_002 | With noblock signature the IUT can raise exception | Clause 14 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_1400_procedure_signatures_003 | Non blocking signatures can raise exception | Clause 14 | m | y |
| 5 | Sem_1400_procedure_signatures_004 | Multiple calls can be send without ack using non-blocking signature | Clause 14 | m | y |

# 5.80. Declaring templates

*Table 79. Declaring templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_15_TopLevel_001 | A template formed from a union is rejected when the union somehow contains a default type field. | Clause 15 | m | n |
| 2 | NegSem_15_TopLevel_002 | A template formed from a union is rejected when the union somehow contains a port type field. | Clause 15 | m | n |
| 3 | NegSem_15_TopLevel_003 | A template shall not be of default type. | Clause 15 | m | n |
| 4 | NegSem_15_TopLevel_004 | A template shall not be of port type. | Clause 15 | m | n |
| 5 | NegSyn_15_TopLevel_001 | The expression or template body initializing a template shall evaluate to a value or template, which is type compatible with the template being declared | Clause 15 | m | y |
| 6 | Syn_15_TopLevel_001 | A simple template with a single charstring field is accepted. | Clause 15 | m | y |

# 5.81. Declaring message templates

*Table 80. Declaring message templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_1501_DeclaringMessageTemplates_001 | A simple record-based message template can be defined. | Clause 15.1 | m | y |
| 2 | Syn_1501_DeclaringMessageTemplates_002 | A simple record-based message template with a wildcard ? is accepted. | Clause 15.1 | m | y |
| 3 | Syn_1501_DeclaringMessageTemplates_003 | A simple record-based message template can be defined with a pattern in a charstring field. | Clause 15.1 | m | y |
| 4 | Syn_1501_DeclaringMessageTemplates_004 | A primitive type template can be defined with a ? wildcard. | Clause 15.1 | m | y |
| 5 | Syn_1501_DeclaringMessageTemplates_005 | A primitive type template can be defined with a one-of notation. | Clause 15.1 | m | y |
| 6 | Syn_1501_DeclaringMessageTemplates_006 | All port operations are accepted. | Clause 15.1 | m | y |

# 5.82. Declaring signature templates

*Table 81. Declaring signature templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_1502_DeclaringSignatureTemplates_001 | Test in-line templates for accepting procedure replies. | Clause 15.2 | m | y |
| 2 | Sem_1502_DeclaringSignatureTemplates_002 | Test in-line templates for accepting procedure replies. | Clause 15.2 | m | y |
| 3 | Sem_1502_DeclaringSignatureTemplates_003 | Test in-line templates for accepting procedure replies. | Clause 15.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Syn_1502_DeclaringSignatureTemplates_001 | Signature templates with explicit values are accepted. | Clause 15.2 | m | y |
| 5 | Syn_1502_DeclaringSignatureTemplates_002 | Signature templates with wildcards are accepted. | Clause 15.2 | m | y |
| 6 | Syn_1502_DeclaringSignatureTemplates_003 | The basic operations `call` and `getreply` are accepted. | Clause 15.2 | m | y |
| 7 | Syn_1502_DeclaringSignatureTemplates_004 | The `raise` and `catch` operations are accepted. | Clause 15.2 | m | y |

# 5.83. Global and local templates

*Table 82. Global and local templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1503_GlobalAndLocalTemplates_001 | There's an error for re-assignment of a global non-parameterized template | Clause 15.3 | m | y |
| 2 | NegSem_1503_GlobalAndLocalTemplates_002 | There's an error for re-assignment of a global non-parameterized template | Clause 15.3 | m | y |
| 3 | NegSem_1503_GlobalAndLocalTemplates_003 | There's an error for re-assignment of a global parameterized template | Clause 15.3 | m | y |
| 4 | NegSem_1503_GlobalAndLocalTemplates_004 | There's an error for re-assignment of a local parameterized template | Clause 15.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSyn_1503_GlobalAndLocalTemplates_001 | There's an error if no value is assigned in a global non-parameterized template declaration | Clause 15.3 | m | y |
| 6 | NegSyn_1503_GlobalAndLocalTemplates_002 | There's an error if no value is assigned in a local non-parameterized template declaration | Clause 15.3 | m | y |
| 7 | NegSyn_1503_GlobalAndLocalTemplates_003 | There's an error if no value is assigned in a global parameterized template declaration | Clause 15.3 | m | y |
| 8 | NegSyn_1503_GlobalAndLocalTemplates_004 | There's an error if no value is assigned in a local parameterized template declaration | Clause 15.3 | m | y |
| 9 | Sem_1503_GlobalAndLocalTemplates_001 | A template values can be accessed with the dot notation as expected. | Clause 15.3 | m | y |
| 10 | Sem_1503_GlobalAndLocalTemplates_002 | A template actual parameter is passed through correctly. | Clause 15.3 | m | y |
| 11 | Sem_1503_GlobalAndLocalTemplates_003 | A `send` operation with actual parameters of a global parameterized template is accepted. | Clause 15.3 | m | y |
| 12 | Sem_1503_GlobalAndLocalTemplates_004 | A parameterized local template in a test case is accepted. | Clause 15.3 | m | n |
| 13 | Sem_1503_GlobalAndLocalTemplates_005 | A `send` operation with actual parameters of a global parameterized template is accepted with the actual parameter being a template parameter. | Clause 15.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_1503_GlobalAndLocalTemplates_006 | A `send` operation with actual parameters of a global parameterized template is accepted with the actual parameter being an inline template. | Clause 15.3 | m | y |
| 15 | Syn_1503_GlobalAndLocalTemplates_001 | A global parameterized template is accepted. | Clause 15.3 | m | y |
| 16 | Syn_1503_GlobalAndLocalTemplates_004 | A parameterized local template in the control part is accepted. | Clause 15.3 | m | n |
| 17 | Syn_1503_GlobalAndLocalTemplates_005 | A parameterized local template in a function is accepted. | Clause 15.3 | m | n |
| 18 | Syn_1503_GlobalAndLocalTemplates_006 | A parameterized local template in an altstep is accepted. | Clause 15.3 | m | n |

# 5.84. In-line templates

*Table 83. In-line templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_1504_InlineTemplates_001 | Inline templates are accepted. | Clause 15.4 | m | y |
| 2 | Syn_1504_InlineTemplates_002 | Modified parameterized inline templates are accepted. | Clause 15.4 | m | y |
| 3 | Syn_1504_InlineTemplates_003 | Modified plain inline templates are accepted. | Clause 15.4 | m | y |

# 5.85. Modified templates

*Table 84. Modified templates*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1505_ModifiedTemplates_001 | A modified template does not refer to itself. | Clause 15.5 | m | y |
| 2 | NegSem_1505_ModifiedTemplates_002 | A modified template does not omit possible parameters of the base template. | Clause 15.5 | m | y |
| 3 | NegSem_1505_ModifiedTemplates_003 | A modified template does not omit possible parameters introduced in any modification step. | Clause 15.5 | m | y |
| 4 | NegSem_1505_ModifiedTemplates_004 | Parameter names in modified templates are the same. | Clause 15.5 | m | y |
| 5 | NegSem_1505_ModifiedTemplates_005 | The dash in default parameter values of a modified templates is only accepted when the base template actually has a default value. | Clause 15.5 | m | y |
| 6 | NegSem_1505_ModifiedTemplates_006 | The same parameter name is used when modifying the base template. | Clause 15.5 | m | y |
| 7 | NegSem_1505_ModifiedTemplates_007 | The same parameter type is used when modifying the base template. | Clause 15.5 | m | y |
| 8 | NegSyn_1505_ModifiedTemplates_001 | The base tamplate and modified template cannot be the same | Clause 15.5 | m | y |
| 9 | Sem_1505_ModifiedTemplates_001 | The values of plain modified template definitions are as expected. | Clause 15.5 | m | y |
| 10 | Sem_1505_ModifiedTemplates_002 | A modified template of a record of type using index notation access works as expected. | Clause 15.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_1505_ModifiedTemplates_003 | Default values in formal parameters of modified templates are working as expected. | Clause 15.5 | m | y |
| 12 | Sem_1505_ModifiedTemplates_004 | Default values in formal parameters of modified templates are working as expected when the modified template uses the dash for the default value. | Clause 15.5 | m | y |
| 13 | Sem_1505_ModifiedTemplates_005 | Default values in formal parameters of modified templates are working as expected | Clause 15.5 | m | y |
| 14 | Sem_1505_ModifiedTemplates_006 | Default values in formal parameters of modified templates are working as expected | Clause 15.5 | m | y |
| 15 | Sem_1505_ModifiedTemplates_007 | Default values in formal parameters of modified templates are working as expected. | Clause 15.5 | m | y |
| 16 | Sem_1505_ModifiedTemplates_008 | The values of plain modified template definitions are as expected. | Clause 15.5 | m | y |
| 17 | Sem_1505_ModifiedTemplates_009 | Default values in formal parameters of modified templates are working as expected. | Clause 15.5 | m | y |
| 18 | Sem_1505_ModifiedTemplates_010 | Default values in formal parameters of modified templates are working as expected. | Clause 15.5 | m | y |
| 19 | Syn_1505_ModifiedTemplates_001 | Plain modified template definitions are accepted. | Clause 15.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 20 | Syn_1505_ModifiedTemplates_002 | A modified template does not omit possible parameters introduced in any modification step. | Clause 15.5 | m | y |
| 21 | Syn_1505_ModifiedTemplates_003 | The default values in formal parameters of modified templates are accepted. | Clause 15.5 | m | y |
| 22 | Syn_1505_ModifiedTemplates_004 | Dash as default parameter values are accepted. | Clause 15.5 | m | y |

# 5.86. Referencing individual string elements

*Table 85. Referencing individual string elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_150601_ReferencingIndividualStringElements_001 | The referencing of individual string elements inside templates or template fields is forbidden. | Clause 15.6.1 | m | y |

# 5.87. Referencing record and set fields

*Table 86. Referencing record and set fields*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_150602_ReferencingRecordAndSetFields_001 | Fields with omit values on the right-hand side of an assignment are rejected. | Clause 15.6.2 | m | y |
| 2 | NegSem_150602_ReferencingRecordAndSetFields_002 | Fields with * values on the right-hand side of an assignment are rejected | Clause 15.6.2 | m | n |
| 3 | NegSem_150602_ReferencingRecordAndSetFields_003 | Value lists on the right-hand side of an assignment are not accepted. | Clause 15.6.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_150602_ReferencingRecordAndSetFields_004 | Complement lists on the right-hand side of an assignment are not accepted. | Clause 15.6.2 | m | y |
| 5 | NegSem_150602_ReferencingRecordAndSetFields_005 | Referencing a template field with the `ifpresent` attribute causes a rejection. | Clause 15.6.2 | m | y |
| 6 | NegSem_150602_ReferencingRecordAndSetFields_006 | Referencing a field of an address type, which actual value is `null` shall cause rejection. | Clause 15.6.2 | m | n |
| 7 | Sem_150602_ReferencingRecordAndSetFields_001 | ? shall be returned for mandatory subfields and * shall be returned for optional subfields. | Clause 15.6.2 | m | y |
| 8 | Sem_150602_ReferencingRecordAndSetFields_002 | The recurisve anyvalue expansion is performed correctly when new values are assigned. | Clause 15.6.2 | m | y |
| 9 | Sem_150602_ReferencingRecordAndSetFields_003 | ? shall be returned for mandatory subfields and * shall be returned for optional subfields. | Clause 15.6.2 | m | n |
| 10 | Sem_150602_ReferencingRecordAndSetFields_004 | ? shall be returned for mandatory subfields and * shall be returned for optional subfields. | Clause 15.6.2 | m | n |

# 5.88. Referencing record of and set of elements

*Table 87. Referencing record of and set of elements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_150603_ReferencingRecordOfAndSetElements_001 | Referencing an element within a value list causes an error in the context of record of. | Clause 15.6.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_150603_ReferencingRecordOfAndSetElements_002 | Access to unitialized fields in the context of record of is rejected. | Clause 15.6.3 | m | y |
| 3 | NegSem_150603_ReferencingRecordOfAndSetElements_003 | Anyvalueornone fields in the context of record of is rejected. | Clause 15.6.3 | m | y |
| 4 | NegSem_150603_ReferencingRecordOfAndSetElements_004 | Complement value lists in the context of record of are rejected. | Clause 15.6.3 | m | y |
| 5 | NegSem_150603_ReferencingRecordOfAndSetElements_005 | Subset in the context of record of are rejected. | Clause 15.6.3 | m | y |
| 6 | NegSem_150603_ReferencingRecordOfAndSetElements_006 | Superset in the context of record of are rejected. | Clause 15.6.3 | m | y |
| 7 | NegSem_150603_ReferencingRecordOfAndSetElements_007 | Access into permutation in record of templates is forbidden. | Clause 15.6.3 | m | n |
| 8 | NegSem_150603_ReferencingRecordOfAndSetElements_008 | Access to record of indexes is forbidden when a previous index entry is a permutation with a *. | Clause 15.6.3 | m | y |
| 9 | NegSem_150603_ReferencingRecordOfAndSetElements_009 | Access to ifpresent fields is not allowed. | Clause 15.6.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 10 | NegSem_150603_ReferencingRecordOfAndSetElements_010 | Referencing AnyValueOrNone fields is not allowed. | Clause 15.6.3 | m | y |
| 11 | NegSem_150603_ReferencingRecordOfAndSetElements_011 | Referencing uninitialized fields is not allowed. | Clause 15.6.3 | m | y |
| 12 | NegSem_150603_ReferencingRecordOfAndSetElements_012 | Referencing uninitialized fields is not allowed. | Clause 15.6.3 | m | y |
| 13 | NegSem_150603_ReferencingRecordOfAndSetElements_013 | Referencing uninitialized fields is not allowed. | Clause 15.6.3 | m | y |
| 14 | NegSem_150603_ReferencingRecordOfAndSetElements_014 | Referencing an element within a value list causes an error in the context of set of. | Clause 15.6.3 | m | y |
| 15 | NegSem_150603_ReferencingRecordOfAndSetElements_015 | Referencing an element of an address type, which actual value is `null` shall cause an error. | Clause 15.6.3 | m | n |
| 16 | Sem_150603_ReferencingRecordOfAndSetElements_001 | Assignment of an anyvalue on the right hand side yields an anyvalue in the context of record of. | Clause 15.6.3 | m | y |
| 17 | Sem_150603_ReferencingRecordOfAndSetElements_002 | Assignment to a anyvalue in the context of record of is handled correctly. | Clause 15.6.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | Sem_150603_ReferencingRecordOfAndSetElements_003 | Assignment to a anyvalue in the context of record of is handled correctly in two subsequent assignments. | Clause 15.6.3 | m | n |
| 19 | Sem_150603_ReferencingRecordOfAndSetElements_004 | Assignment to a anyvalue in the context of record of is handled correctly when the first element is changed. | Clause 15.6.3 | m | y |
| 20 | Sem_150603_ReferencingRecordOfAndSetElements_005 | Access outside permutation fields is allowed and works as expected. | Clause 15.6.3 | m | y |
| 21 | Sem_150603_ReferencingRecordOfAndSetElements_006 | Referencing an element within a record of, set of or array field to which omit is assigned works as expected | Clause 15.6.3 | m | y |
| 22 | Sem_150603_ReferencingRecordOfAndSetElements_007 | Referencing an element within a record of, set of or array field to which omit is assigned works as expected | Clause 15.6.3 | m | n |

# 5.89. Referencing signature parameters

*Table 88. Referencing signature parameters*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_150604_ReferencingSignatureParameters_001 | Test modification of signature parameters. | Clause 15.6.4 | m | n |
| 2 | Sem_150604_ReferencingSignatureParameters_001 | Test modification of signature parameters. | Clause 15.6.4 | m | y |

# 5.90. Referencing union alternatives

*Table 89. Referencing union alternatives*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_150605_Referencing_union_alternatives_001 | Template variables does not allow referencing alternatives inside an union with omit | Clause 15.6.5 | m | y |
| 2 | NegSem_150605_Referencing_union_alternatives_002 | Template variables does not allow referencing alternatives inside an union with AnyValueOrNone | Clause 15.6.5 | m | y |
| 3 | NegSem_150605_Referencing_union_alternatives_003 | Template variables does not allow referencing alternatives inside an union with list | Clause 15.6.5 | m | n |
| 4 | NegSem_150605_Referencing_union_alternatives_004 | Template variables does not allow referencing alternatives inside an union with complemented list | Clause 15.6.5 | m | n |
| 5 | NegSem_150605_Referencing_union_alternatives_005 | Referencing an alternative of a union template field to which the `ifpresent` attribute is attached, shall cause an error | Clause 15.6.5 | m | n |
| 6 | NegSem_150605_Referencing_union_alternatives_006 | Referencing an alternative of an address type, which actual value is `null` shall cause | Clause 15.6.5 | m | n |
| 7 | Sem_150605_Referencing_union_alternatives_001 | Template variables allow referencing alternatives inside a union template definition | Clause 15.6.5 | m | y |
| 8 | Sem_150605_Referencing_union_alternatives_002 | Template variables allow referencing with an Anyvalue union template | Clause 15.6.5 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | Sem_150605_Referencing_union_alternatives_003 | Template variables allow referencing with an Anyvalue union template | Clause 15.6.5 | m | y |
| 10 | Sem_150605_Referencing_union_alternatives_004 | Template variables allow referencing with an Anyvalue union template | Clause 15.6.5 | m | y |

# 5.91. Template restrictions

*Table 90. Template restrictions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1508_TemplateRestrictions_001 | Template(omit) is rejected with anyvalue(?). | Clause 15.8 | m | y |
| 2 | NegSem_1508_TemplateRestrictions_002 | Template(omit) is rejected with setof template. | Clause 15.8 | m | y |
| 3 | NegSem_1508_TemplateRestrictions_003 | Template(omit) is rejected with anyvalueornone(*). | Clause 15.8 | m | y |
| 4 | NegSem_1508_TemplateRestrictions_004 | Template(omit) is rejected with value ranges. | Clause 15.8 | m | y |
| 5 | NegSem_1508_TemplateRestrictions_005 | Template(omit) is rejected with supersets. | Clause 15.8 | m | y |
| 6 | NegSem_1508_TemplateRestrictions_006 | Template(omit) is rejected with subsets. | Clause 15.8 | m | y |
| 7 | NegSem_1508_TemplateRestrictions_007 | Template(omit) is rejected with patterns. | Clause 15.8 | m | y |
| 8 | NegSem_1508_TemplateRestrictions_008 | Template(omit) is rejected with anyelement inside values. | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_1508_TemplateRestrictions_009 | Template(omit) is rejected with anyelemenornone inside values. | Clause 15.8 | m | y |
| 10 | NegSem_1508_TemplateRestrictions_010 | Template(omit) is rejected with permutation inside values. | Clause 15.8 | m | y |
| 11 | NegSem_1508_TemplateRestrictions_011 | Template(omit) is rejected with length restrictions. | Clause 15.8 | m | y |
| 12 | NegSem_1508_TemplateRestrictions_012 | Template(omit) is rejected with length restrictions. | Clause 15.8 | m | y |
| 13 | NegSem_1508_TemplateRestrictions_013 | Template(omit) is rejected with length restrictions. | Clause 15.8 | m | y |
| 14 | NegSem_1508_TemplateRestrictions_014 | Template(value) is rejected with anyvalue(?). | Clause 15.8 | m | y |
| 15 | NegSem_1508_TemplateRestrictions_015 | Template(value) is rejected with valuelist. | Clause 15.8 | m | y |
| 16 | NegSem_1508_TemplateRestrictions_016 | Template(value) is rejected with anyvalueornone(*). | Clause 15.8 | m | y |
| 17 | NegSem_1508_TemplateRestrictions_017 | Template(value) is rejected with value ranges. | Clause 15.8 | m | y |
| 18 | NegSem_1508_TemplateRestrictions_018 | Template(value) is rejected with supersets. | Clause 15.8 | m | y |
| 19 | NegSem_1508_TemplateRestrictions_019 | Template(value) is rejected with supersets. | Clause 15.8 | m | y |
| 20 | NegSem_1508_TemplateRestrictions_020 | Template(value) is rejected with patterns. | Clause 15.8 | m | y |
| 21 | NegSem_1508_TemplateRestrictions_021 | Template(value) is rejected with anyelement inside values. | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 22 | NegSem_1508_TemplateRestrictions_022 | Template(value) is rejected with permutation inside values. | Clause 15.8 | m | y |
| 23 | NegSem_1508_TemplateRestrictions_023 | Template(value) is rejected with length restrictions. | Clause 15.8 | m | y |
| 24 | NegSem_1508_TemplateRestrictions_024 | Template(value) is rejected with length restrictions. | Clause 15.8 | m | y |
| 25 | NegSem_1508_TemplateRestrictions_025 | Template(present) refuses omit value as a whole. | Clause 15.8 | m | y |
| 26 | NegSem_1508_TemplateRestrictions_026 | Template(value) refuses omit as a whole. | Clause 15.8 | m | y |
| 27 | NegSem_1508_TemplateRestrictions_027 | ensure that symbols created during template expansion are checked against omit template restriction | Clause 15.8 | m | n |
| 28 | NegSem_1508_TemplateRestrictions_028 | ensure that symbols created during template expansion are checked against value template restriction | Clause 15.8 | m | n |
| 29 | NegSem_1508_TemplateRestrictions_029 | The template(present) with anyvalue(?) can’t be assigned to an omit restricted variable template | Clause 15.8 | m | y |
| 30 | NegSem_1508_TemplateRestrictions_030 | Unrestricted template with anyvalue(?) can’t be assigned to an omit restricted variable template | Clause 15.8 | m | y |
| 31 | NegSem_1508_TemplateRestrictions_031 | Template(omit) can’t be assigned to a variable template(value) if omit | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 32 | NegSem_1508_TemplateRestrictions_032 | Template(present) can't be assigned to a template(value) variable if contains anyvalueornone(*) | Clause 15.8 | m | y |
| 33 | NegSem_1508_TemplateRestrictions_033 | An unrestricted template can't be assigned to a template(value) variable if contains anyvalueornone(*) | Clause 15.8 | m | y |
| 34 | NegSem_1508_TemplateRestrictions_034 | A template with omit restriction can't be assigned to a template(present)variable if omit | Clause 15.8 | m | y |
| 35 | NegSem_1508_TemplateRestrictions_035 | An unrestricted template can't be assigned to a template(present)variable if omit | Clause 15.8 | m | y |
| 36 | NegSem_1508_TemplateRestrictions_036 | Template(present) can't be parameter to a template(omit) if contains anyvalueornone(*) | Clause 15.8 | m | y |
| 37 | NegSem_1508_TemplateRestrictions_037 | Template(present) can't be parameter to template(omit) if contains anyvalue(?) | Clause 15.8 | m | y |
| 38 | NegSem_1508_TemplateRestrictions_038 | Template(present) can't be parameter to template(value) if it contains anyvalueornone(*) | Clause 15.8 | m | y |
| 39 | NegSem_1508_TemplateRestrictions_039 | Unrestricted template can't be parameter to template(value) if it contains anyvalueornone(*) | Clause 15.8 | m | y |
| 40 | NegSem_1508_TemplateRestrictions_040 | Template(present) can't be parameter to a template(omit) | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 41 | NegSem_1508_TemplateRestrictions_041 | Unrestricted template cannot be parameter to template(value) | Clause 15.8 | m | y |
| 42 | NegSem_1508_TemplateRestrictions_042 | Template(present) cannot be parameter to template(value) | Clause 15.8 | m | y |
| 43 | NegSem_1508_TemplateRestrictions_043 | Template(present) cannot be parameter to template(omit) | Clause 15.8 | m | y |
| 44 | NegSem_1508_TemplateRestrictions_044 | The restrictiveness of parameters template(value)→template(present) is handled correctly. | Clause 15.8 | m | y |
| 45 | NegSem_1508_TemplateRestrictions_045 | The restrictiveness of parameters template(value)→template(omit) is handled correctly. | Clause 15.8 | m | y |
| 46 | NegSem_1508_TemplateRestrictions_046 | The restrictiveness of parameters template(value)→template is handled correctly. | Clause 15.8 | m | y |
| 47 | NegSem_1508_TemplateRestrictions_047 | The restrictiveness of parameters template(omit)→template(present) is handled correctly. | Clause 15.8 | m | y |
| 48 | NegSem_1508_TemplateRestrictions_048 | The restrictiveness of parameters template(omit)→template(present) is handled correctly. | Clause 15.8 | m | y |
| 49 | NegSem_1508_TemplateRestrictions_049 | The restrictiveness of parameters template(omit)→template(present) is handled correctly. | Clause 15.8 | m | y |
| 50 | NegSem_1508_TemplateRestrictions_050 | Decoded content match is not allowed for omit template restriction | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 51 | NegSem_1508_TemplateRestrictions_051 | Decoded content match is not allowed for omit template restriction | Clause 15.8 | m | y |
| 52 | Sem_1508_TemplateRestrictions_001 | A value can be assigned to a template(omit) variable. | Clause 15.8 | m | y |
| 53 | Sem_1508_TemplateRestrictions_002 | A template(omit) can be assigned to a template(omit) variable. | Clause 15.8 | m | y |
| 54 | Sem_1508_TemplateRestrictions_003 | A template(value) can be assigned to a template(omit) variable. | Clause 15.8 | m | y |
| 55 | Sem_1508_TemplateRestrictions_004 | A value can be assigned to a template(value) variable. | Clause 15.8 | m | y |
| 56 | Sem_1508_TemplateRestrictions_005 | A template(value) can be assigned to a template(value) variable. | Clause 15.8 | m | y |
| 57 | Sem_1508_TemplateRestrictions_006 | A value can be assigned to a template(present) variable. | Clause 15.8 | m | y |
| 58 | Sem_1508_TemplateRestrictions_007 | A template(omit) can be assigned to a template(present) variable. | Clause 15.8 | m | y |
| 59 | Sem_1508_TemplateRestrictions_008 | A template(value) can be assigned to a template(present) variable. | Clause 15.8 | m | y |
| 60 | Sem_1508_TemplateRestrictions_009 | A template(present) can be assigned to a template(present) variable. | Clause 15.8 | m | y |
| 61 | Sem_1508_TemplateRestrictions_010 | A value can be assigned to a template variable. | Clause 15.8 | m | y |
| 62 | Sem_1508_TemplateRestrictions_011 | A template(omit) can be assigned to a template variable. | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 63 | Sem_1508_TemplateRestrictions_012 | A template(value) can be assigned to a template variable. | Clause 15.8 | m | y |
| 64 | Sem_1508_TemplateRestrictions_013 | A template(present) can be assigned to a template variable. | Clause 15.8 | m | y |
| 65 | Sem_1508_TemplateRestrictions_014 | A template can be assigned to a template variable. | Clause 15.8 | m | y |
| 66 | Sem_1508_TemplateRestrictions_015 | A base template can be modified without restrictions. | Clause 15.8 | m | y |
| 67 | Sem_1508_TemplateRestrictions_016 | A base template can be modified with template(present) restriction. | Clause 15.8 | m | y |
| 68 | Sem_1508_TemplateRestrictions_017 | A base template can be modified with template(omit) restriction. | Clause 15.8 | m | y |
| 69 | Sem_1508_TemplateRestrictions_018 | A base template can be modified with template(value) restriction. | Clause 15.8 | m | y |
| 70 | Sem_1508_TemplateRestrictions_019 | A template(present) base template can be modified with template(present) restriction. | Clause 15.8 | m | y |
| 71 | Sem_1508_TemplateRestrictions_020 | A template(present) base template can be modified with template(value) restriction. | Clause 15.8 | m | y |
| 72 | Sem_1508_TemplateRestrictions_021 | A template(omit) base template can be modified with template(omit) restriction. | Clause 15.8 | m | y |
| 73 | Sem_1508_TemplateRestrictions_022 | A template(omit) base template can be modified with template(value) restriction. | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 74 | Sem_1508_TemplateRestrictions_023 | A template(value) base template can be modified with template(value) restriction. | Clause 15.8 | m | y |
| 75 | Sem_1508_TemplateRestrictions_024 | Template(present) base templates are allowed to be modified to template(omit). | Clause 15.8 | m | y |
| 76 | Sem_1508_TemplateRestrictions_025 | Template(omit) base templates are allowed to be modified to template(present). | Clause 15.8 | m | y |
| 77 | Sem_1508_TemplateRestrictions_026 | Template(value) base templates are allowed to be modified to template(present). | Clause 15.8 | m | y |
| 78 | Sem_1508_TemplateRestrictions_027 | Template(value) base templates are allowed to be modified to template(omit). | Clause 15.8 | m | y |
| 79 | Sem_1508_TemplateRestrictions_028 | Template(value) base templates are allowed to be modified to template. | Clause 15.8 | m | y |
| 80 | Sem_1508_TemplateRestrictions_029 | Template(omit) base templates are allowed to be modified to template. | Clause 15.8 | m | y |
| 81 | Sem_1508_TemplateRestrictions_030 | Template(present) base templates are allowed to be modified to template. | Clause 15.8 | m | y |
| 82 | Sem_1508_TemplateRestrictions_031 | Template (omit) can be parameter to template(present) if it contains omit | Clause 15.8 | m | y |
| 83 | Sem_1508_TemplateRestrictions_032 | An unrestricted template can’t be parameter to template(present) if it contains omit | Clause 15.8 | m | y |
| 84 | Sem_1508_TemplateRestrictions_033 | An unrestricted template can be parameter to template(present) | Clause 15.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 85 | Sem_1508_TemplateRestrictions_034 | Template (omit) can be parameter to template(present) | Clause 15.8 | m | y |
| 86 | Sem_1508_TemplateRestrictions_035 | Template(omit) can be parameter to template(value) if it is omit | Clause 15.8 | m | y |
| 87 | Sem_1508_TemplateRestrictions_036 | Template(omit) can be parameter to template(value) | Clause 15.8 | m | y |
| 88 | Sem_1508_TemplateRestrictions_037 | Decoded content match is allowed for present template restriction | Clause 15.8 | m | y |
| 89 | Syn_1508_TemplateRestrictions_001 | Template(omit) is accepted with value omit value. | Clause 15.8 | m | y |
| 90 | Syn_1508_TemplateRestrictions_002 | Template(omit) is accepted with a concrete value. | Clause 15.8 | m | y |
| 91 | Syn_1508_TemplateRestrictions_003 | Template(value) is accepted with a concrete value. | Clause 15.8 | m | y |
| 92 | Syn_1508_TemplateRestrictions_004 | Template(present) is accepted with a concrete value. | Clause 15.8 | m | y |

## 5.92. `Match` operation

*Table 91. `Match` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1509_MatchOperation_001 | The `match` operation refuses two templates as actual parameters. | Clause 15.9 | m | y |
| 2 | NegSem_1509_MatchOperation_002 | The `match` operation refuses not initialized operands | Clause 15.9 | m | n |
| 3 | NegSem_1509_MatchOperation_003 | The `match` operation works correctly with enums | Clause 15.9 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_1509_MatchOperation_001 | The `match` operation works as expected on a template with range restriction when the tested value is inside the range. | Clause 15.9 | m | y |
| 5 | Sem_1509_MatchOperation_002 | The `match` operation works as expected on a template with range restriction when the tested value is outside the range. | Clause 15.9 | m | y |
| 6 | Sem_1509_MatchOperation_003 | The `match` operation works correctly on records in the positive case. | Clause 15.9 | m | y |
| 7 | Sem_1509_MatchOperation_004 | The `match` operation works correctly on records in the negative case. | Clause 15.9 | m | y |
| 8 | Sem_1509_MatchOperation_005 | The `match` operation works correctly if the types are incompatible. | Clause 15.9 | m | n |
| 9 | Sem_1509_MatchOperation_006 | The `match` operation works correctly on records with optional fields in the positive case. | Clause 15.9 | m | y |
| 10 | Sem_1509_MatchOperation_007 | The `match` operation works correctly on sets in the positive case. | Clause 15.9 | m | y |
| 11 | Sem_1509_MatchOperation_008 | The `match` operation works correctly on sets in the negative case. | Clause 15.9 | m | y |
| 12 | Sem_1509_MatchOperation_009 | The `match` operation works correctly if the set types are incompatible. | Clause 15.9 | m | n |
| 13 | Sem_1509_MatchOperation_010 | The `match` operation works correctly on sets with optional fields in the positive case. | Clause 15.9 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_1509_MatchOperation_011 | Matching a value expression against a template instance which evaluates to the omit matching mechanism shall return false | Clause 15.9 | m | y |
| 15 | Sem_1509_MatchOperation_012 | If the `expression` -parameter evaluates to a literal value without explicit or implicit identification of its type, the type of the template `instance`-parameter shall be used as the type governor for the `expression`-parameter. | Clause 15.9 | m | y |
| 16 | Sem_1509_MatchOperation_013 | If the `expression` -parameter evaluates to a literal value without explicit or implicit identification of its type, the type of the template instance-parameter shall be used as the type governor for the `expression`-parameter | Clause 15.9 | m | y |
| 17 | Sem_1509_MatchOperation_014 | The `match` operation works correctly with enums | Clause 15.9 | m | n |
| 18 | Sem_1509_MatchOperation_015 | The `match` operation works correctly with enums | Clause 15.9 | m | y |
| 19 | Sem_1509_MatchOperation_016 | The `match` operation works correctly with enums | Clause 15.9 | m | y |

## 5.93. `Valueof` operation

*Table 92. `Valueof` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1510_ValueOfOperation_001 | The `valueof` function works correctly on omit. | Clause 15.10 | m | y |
| 2 | NegSem_1510_ValueOfOperation_002 | The `valueof` function works correctly on templates with wildcards. | Clause 15.10 | m | y |
| 3 | NegSem_1510_ValueOfOperation_003 | The `valueof` function works correctly on regular value templates. | Clause 15.10 | m | y |
| 4 | NegSem_1510_ValueOfOperation_004 | The `valueof` function works correctly on range templates. | Clause 15.10 | m | y |
| 5 | NegSem_1510_ValueOfOperation_005 | check that runtime error occurs if `valueof` is applied to uninitialized template | Clause 15.10 | m | y |
| 6 | NegSem_1510_ValueOfOperation_006 | check that runtime error occurs if `valueof` is applied to partially initialized template | Clause 15.10 | m | y |
| 7 | Sem_1510_ValueOfOperation_001 | The `valueof` operation works as expected for fully initialized templates. | Clause 15.10 | m | y |

# 5.94. Concatenating templates of string and list types

*Table 93. Concatenating templates of string and list types*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_001 | Concatenation of octetstring types yields an even number of digits. | Clause 15.11 | m | y |
| 2 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_002 | Concatenation of strings types yields an error if specified ranges are not fixed length. | Clause 15.11 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_003 | A simple concatenation of non-wildcard octetstring must not yield in a non-even number of hexadecimals. | Clause 15.11 | m | y |
| 4 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_004 | The inline template definitions are correctly concatenated. | Clause 15.11 | m | y |
| 5 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_005 | The inline template definitions are correctly concatenated. | Clause 15.11 | m | y |
| 6 | NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_006 | Concatenation of octetstring types and ? patterns works as expected. | Clause 15.11 | m | n |
| 7 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_001 | Concatenation of charstring types works as expected (variant 1). | Clause 15.11 | m | y |
| 8 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_002 | Concatenation of octetstring types works as expected (variant 2). | Clause 15.11 | m | y |
| 9 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_003 | Concatenation of bitstring types works as expected. | Clause 15.11 | m | n |
| 10 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_004 | Concatenation of octetstring types works as expected (variant 1). | Clause 15.11 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_005 | Concatenation of octetstring types works as expected (variant 2). | Clause 15.11 | m | n |
| 12 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_006 | A concatenation of charstrings with a fixed length AnyValueOrNone be matched. | Clause 15.11 | m | y |
| 13 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_007 | Concatenations of record of charstrings are accepted. | Clause 15.11 | m | n |
| 14 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_008 | Concatenations of record of charstrings work when parameterized. | Clause 15.11 | m | n |
| 15 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_009 | Concatenations of set of integers are accepted. | Clause 15.11 | m | n |
| 16 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_010 | The inline template definitions are correctly concatenated. | Clause 15.11 | m | y |
| 17 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_011 | Concatenation of octetstring types works as expected (matching patterns in quotation). | Clause 15.11 | m | n |
| 18 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_012 | Concatenation of octetstring types and ? patterns works as expected. | Clause 15.11 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_013 | Concatenation of octetstring types and ? patterns works as expected. | Clause 15.11 | m | y |
| 20 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_014 | Concatenation of charstring and universal charsting types are concatenated as expected. | Clause 15.11 | m | y |
| 21 | Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_015 | Concatenations of record of charstrings work when parameterized | Clause 15.11 | m | n |

# 5.95. Functions

*Table 94. Functions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1601_toplevel_001 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 2 | NegSem_1601_toplevel_002 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 3 | NegSem_1601_toplevel_003 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 4 | NegSem_1601_toplevel_004 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 5 | NegSem_1601_toplevel_005 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 6 | NegSem_1601_toplevel_006 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 7 | Sem_1601_toplevel_001 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 8 | Sem_1601_toplevel_002 | The IUT correctly handles function definitions | Clause 16.1 | m | y |
| 9 | Sem_1601_toplevel_003 | The IUT correctly handles function definitions | Clause 16.1 | m | y |

# 5.96. Invoking functions

*Table 95. Invoking functions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_160101_invoking_functions_001 | The IUT correctly handles function invocations | Clause 16.1.1 | m | y |

# 5.97. Predefined functions

*Table 96. Predefined functions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_160102_predefined_functions_001 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 2 | NegSem_160102_predefined_functions_002 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 3 | NegSem_160102_predefined_functions_003 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 4 | NegSem_160102_predefined_functions_004 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 5 | NegSem_160102_predefined_functions_005 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 6 | NegSem_160102_predefined_functions_006 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 7 | NegSem_160102_predefined_functions_007 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_160102_predefined_functions_008 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 9 | NegSem_160102_predefined_functions_009 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 10 | NegSem_160102_predefined_functions_010 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 11 | NegSem_160102_predefined_functions_017 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 12 | NegSem_160102_predefined_functions_018 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 13 | NegSem_160102_predefined_functions_019 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 14 | NegSem_160102_predefined_functions_021 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 15 | NegSem_160102_predefined_functions_022 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 16 | NegSem_160102_predefined_functions_023 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 17 | NegSem_160102_predefined_functions_024 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | NegSem_160102_predefined_functions_025 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 19 | NegSem_160102_predefined_functions_026 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 20 | NegSem_160102_predefined_functions_027 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 21 | NegSem_160102_predefined_functions_028 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 22 | NegSem_160102_predefined_functions_029 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 23 | NegSem_160102_predefined_functions_030 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 24 | NegSem_160102_predefined_functions_031 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 25 | NegSem_160102_predefined_functions_032 | An error is generated when the parameter of the `encvalue` function contains a matching symbol | Clause 16.1.2 | m | y |
| 26 | NegSem_160102_predefined_functions_033 | An error is detected when the parameter of the `encvalue` function contains an unitialized value | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 27 | NegSem_160102_predefined_functions_034 | An error is detected when the parameter of the `encvalue` function contains a partially initialized value | Clause 16.1.2 | m | y |
| 28 | NegSem_160102_predefined_functions_035 | An error is detected when the first parameter of the `decvalue` function contains an uninitialized value | Clause 16.1.2 | m | y |
| 29 | NegSem_160102_predefined_functions_036 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 30 | NegSem_160102_predefined_functions_037 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 31 | NegSem_160102_predefined_functions_038 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 32 | NegSem_160102_predefined_functions_039 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 33 | NegSem_160102_predefined_functions_040 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 34 | Sem_160102_predefined_functions_001 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 35 | Sem_160102_predefined_functions_002 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 36 | Sem_160102_predefined_functions_003 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 37 | Sem_160102_predefined_functions_004 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 38 | Sem_160102_predefined_functions_005 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 39 | Sem_160102_predefined_functions_006 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 40 | Sem_160102_predefined_functions_007 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 41 | Sem_160102_predefined_functions_008 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 42 | Sem_160102_predefined_functions_009 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 43 | Sem_160102_predefined_functions_010 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 44 | Sem_160102_predefined_functions_011 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 45 | Sem_160102_predefined_functions_012 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 46 | Sem_160102_predefined_functions_013 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 47 | Sem_160102_predefined_functions_014 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 48 | Sem_160102_predefined_functions_015 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 49 | Sem_160102_predefined_functions_016 | Predefined `encvalue` function works correctly (as specified in Annex C.5.1) | Clause 16.1.2 | m | y |
| 50 | Sem_160102_predefined_functions_017 | Predefined `decvalue` function performs full decoding correctly | Clause 16.1.2 | m | y |
| 51 | Sem_160102_predefined_functions_018 | Predefined `decvalue` function performs decoding if there are more bits than needed | Clause 16.1.2 | m | y |
| 52 | Sem_160102_predefined_functions_019 | Predefined `decvalue` function works properly in case of decoding failure | Clause 16.1.2 | m | n |
| 53 | Sem_160102_predefined_functions_020 | Predefined `decvalue` function works properly in case of not enough bits | Clause 16.1.2 | m | n |
| 54 | Sem_160102_predefined_functions_021 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 55 | Sem_160102_predefined_functions_022 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 56 | Sem_160102_predefined_functions_023 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 57 | Sem_160102_predefined_functions_024 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 58 | Sem_160102_predefined_functions_025 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.33) | Clause 16.1.2 | m | y |
| 59 | Sem_160102_predefined_functions_026 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 60 | Sem_160102_predefined_functions_027 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 61 | Sem_160102_predefined_functions_028 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 62 | Sem_160102_predefined_functions_029 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 63 | Sem_160102_predefined_functions_030 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 64 | Sem_160102_predefined_functions_031 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 65 | Sem_160102_predefined_functions_032 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 66 | Sem_160102_predefined_functions_033 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 67 | Sem_160102_predefined_functions_034 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 68 | Sem_160102_predefined_functions_035 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 69 | Sem_160102_predefined_functions_036 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 70 | Sem_160102_predefined_functions_037 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 71 | Sem_160102_predefined_functions_038 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 72 | Sem_160102_predefined_functions_039 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 73 | Sem_160102_predefined_functions_040 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 74 | Sem_160102_predefined_functions_041 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 75 | Sem_160102_predefined_functions_042 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 76 | Sem_160102_predefined_functions_043 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 77 | Sem_160102_predefined_functions_044 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 78 | Sem_160102_predefined_functions_045 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 79 | Sem_160102_predefined_functions_046 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 80 | Sem_160102_predefined_functions_047 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 81 | Sem_160102_predefined_functions_048 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 82 | Sem_160102_predefined_functions_049 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 83 | Sem_160102_predefined_functions_050 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 84 | Sem_160102_predefined_functions_051 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 85 | Sem_160102_predefined_functions_052 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 86 | Sem_160102_predefined_functions_053 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 87 | Sem_160102_predefined_functions_054 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 88 | Sem_160102_predefined_functions_055 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 89 | Sem_160102_predefined_functions_056 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | n |
| 90 | Sem_160102_predefined_functions_057 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 91 | Sem_160102_predefined_functions_058 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 92 | Sem_160102_predefined_functions_059 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 93 | Sem_160102_predefined_functions_060 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 94 | Sem_160102_predefined_functions_061 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 95 | Sem_160102_predefined_functions_062 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.3.5) | Clause 16.1.2 | m | y |
| 96 | Sem_160102_predefined_functions_063 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 97 | Sem_160102_predefined_functions_064 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 98 | Sem_160102_predefined_functions_065 | Predefined `decvalue` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 99 | Sem_160102_predefined_functions_066 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 100 | Sem_160102_predefined_functions_067 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 101 | Sem_160102_predefined_functions_068 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 102 | Sem_160102_predefined_functions_069 | Predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 103 | Sem_160102_predefined_functions_070 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 104 | Sem_160102_predefined_functions_071 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 105 | Sem_160102_predefined_functions_072 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 106 | Sem_160102_predefined_functions_073 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 107 | Sem_160102_predefined_functions_074 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 108 | Sem_160102_predefined_functions_075 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 109 | Sem_160102_predefined_functions_076 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 110 | Sem_160102_predefined_functions_077 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 111 | Sem_160102_predefined_functions_078 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 112 | Sem_160102_predefined_functions_079 | Predefined `decvalue_unichar` function works properly | Clause 16.1.2 | m | y |
| 113 | Sem_160102_predefined_functions_080 | Predefined `decvalue` and `decvalue_unichar` function works properly in case of uninitialized encode value is given | Clause 16.1.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 114 | Sem_160102_predefined_functions_081 | Predefined function `get_stringencoding` works properly | Clause 16.1.2 | m | y |
| 115 | Sem_160102_predefined_functions_082 | Predefined function for removing Byte order mark works properly | Clause 16.1.2 | m | y |
| 116 | Sem_160102_predefined_functions_083 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 117 | Sem_160102_predefined_functions_084 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 118 | Sem_160102_predefined_functions_085 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | n |
| 119 | Sem_160102_predefined_functions_086 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 120 | Sem_160102_predefined_functions_087 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 121 | Sem_160102_predefined_functions_088 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 122 | Sem_160102_predefined_functions_089 | Predefined function `isvalue()` works properly | Clause 16.1.2 | m | y |
| 123 | Sem_160102_predefined_functions_090 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.4.1) | Clause 16.1.2 | m | y |
| 124 | Sem_160102_predefined_functions_091 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 125 | Sem_160102_predefined_functions_092 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 126 | Sem_160102_predefined_functions_093 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 127 | Sem_160102_predefined_functions_094 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | n |
| 128 | Sem_160102_predefined_functions_095 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 129 | Sem_160102_predefined_functions_096 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 130 | Sem_160102_predefined_functions_097 | The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C) | Clause 16.1.2 | m | y |
| 131 | Sem_160102_predefined_functions_098 | That predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |
| 132 | Sem_160102_predefined_functions_099 | That predefined `encvalue_unichar` function works properly in case of encoding universal charstring | Clause 16.1.2 | m | y |

# 5.98. External functions

*Table 97. External functions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_160103_external_functions_001 | The IUT recognizes external functions | Clause 16.1.3 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_160103_external_functions_001 | The IUT recognizes external functions | Clause 16.1.3 | m | y |
| 3 | Sem_160103_external_functions_002 | The IUT recognizes external functions | Clause 16.1.3 | m | y |

# 5.99. Invoking function from specific places

*Table 98. Invoking function from specific places*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_160104_invoking_functions_from_specific_places_001 | The IUT recognizes restrictions described in section 16.1.4. STF409 assumes that the list given in section 16.1.4 describes mandatory restrictions | Clause 16.1.4 | m | n |
| 2 | NegSem_160104_invoking_functions_from_specific_places_002 | The IUT recognizes restrictions described in section 16.1.4. STF409 assumes that the list given in section 16.1.4 describes mandatory restrictions | Clause 16.1.4 | m | y |
| 3 | NegSem_160104_invoking_functions_from_specific_places_003 | The IUT recognizes restrictions described in section 16.1.4. STF409 assumes that the list given in section 16.1.4 describes mandatory restrictions | Clause 16.1.4 | m | y |
| 4 | NegSem_160104_invoking_functions_from_specific_places_004 | The IUT recognizes restrictions described in section 16.1.4. STF409 assumes that the list given in section 16.1.4 describes mandatory restrictions | Clause 16.1.4 | m | y |
| 5 | NegSem_160104_invoking_functions_from_specific_places_005 | verify that the `create` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_160104_invoking_functions_from_specific_places_006 | verify that the `component.start` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 7 | NegSem_160104_invoking_functions_from_specific_places_007 | verify that the `component.stop` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 8 | NegSem_160104_invoking_functions_from_specific_places_008 | verify that the `kill` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 9 | NegSem_160104_invoking_functions_from_specific_places_009 | verify that the `component.running` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 10 | NegSem_160104_invoking_functions_from_specific_places_010 | verify that the `alive` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 11 | NegSem_160104_invoking_functions_from_specific_places_011 | verify that the `done` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 12 | NegSem_160104_invoking_functions_from_specific_places_012 | verify that the `killed` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | NegSem_160104_invoking_functions_from_specific_places_013 | verify that the `port.start` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 14 | NegSem_160104_invoking_functions_from_specific_places_014 | verify that the `port.stop` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 15 | NegSem_160104_invoking_functions_from_specific_places_015 | verify that the `halt` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 16 | NegSem_160104_invoking_functions_from_specific_places_016 | verify that the `clear` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 17 | NegSem_160104_invoking_functions_from_specific_places_017 | verify that the `checkstate` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 18 | NegSem_160104_invoking_functions_from_specific_places_018 | verify that the `send` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 19 | NegSem_160104_invoking_functions_from_specific_places_019 | verify that the `receive` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 20 | NegSem_160104_invoking_functions_from_specific_places_020 | verify that the `trigger` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 21 | NegSem_160104_invoking_functions_from_specific_places_021 | verify that the `call` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 22 | NegSem_160104_invoking_functions_from_specific_places_022 | verify that the `getcall` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 23 | NegSem_160104_invoking_functions_from_specific_places_023 | verify that the `reply` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 24 | NegSem_160104_invoking_functions_from_specific_places_024 | verify that the `getreply` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 25 | NegSem_160104_invoking_functions_from_specific_places_025 | verify that the `raise` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 26 | NegSem_160104_invoking_functions_from_specific_places_026 | verify that the `catch` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 27 | NegSem_160104_invoking_functions_from_specific_places_027 | verify that the `check` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 28 | NegSem_160104_invoking_functions_from_specific_places_028 | verify that the `connect` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 29 | NegSem_160104_invoking_functions_from_specific_places_029 | verify that the `disconnect` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 30 | NegSem_160104_invoking_functions_from_specific_places_030 | verify that the `map` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 31 | NegSem_160104_invoking_functions_from_specific_places_031 | verify that the `unmap` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 32 | NegSem_160104_invoking_functions_from_specific_places_032 | verify that the `action` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 33 | NegSem_160104_invoking_functions_from_specific_places_033 | verify that the `timer.start` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 34 | NegSem_160104_invoking_functions_from_specific_places_034 | verify that the `timer.stop` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 35 | NegSem_160104_invoking_functions_from_specific_places_035 | verify that the `timer.running` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 36 | NegSem_160104_invoking_functions_from_specific_places_036 | verify that the `read` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 37 | NegSem_160104_invoking_functions_from_specific_places_037 | verify that the `timeout` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 38 | NegSem_160104_invoking_functions_from_specific_places_038 | verify that a non-deterministic external function call cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 39 | NegSem_160104_invoking_functions_from_specific_places_039 | verify that the predefined `rnd` function cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 40 | NegSem_160104_invoking_functions_from_specific_places_040 | verify a function called during receiving operation cannot contain an assignment of a component variable (in templates) | Clause 16.1.4 | m | n |
| 41 | NegSem_160104_invoking_functions_from_specific_places_041 | verify a function called during receiving operation cannot contain a component variable used as an actual `out` parameter (in templates) | Clause 16.1.4 | m | n |
| 42 | NegSem_160104_invoking_functions_from_specific_places_042 | verify a function called during receiving operation cannot contain a component variable used as an actual `inout` parameter (in templates) | Clause 16.1.4 | m | n |
| 43 | NegSem_160104_invoking_functions_from_specific_places_043 | verify that the `setverdict` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 44 | NegSem_160104_invoking_functions_from_specific_places_044 | verify that the `activate` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 45 | NegSem_160104_invoking_functions_from_specific_places_045 | verify that the `deactivate` operation cannot be used in a function called during receiving operation (in templates) | Clause 16.1.4 | m | n |
| 46 | NegSem_160104_invoking_functions_from_specific_places_046 | verify that the `create` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 47 | NegSem_160104_invoking_functions_from_specific_places_047 | verify that the `component.start` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 48 | NegSem_160104_invoking_functions_from_specific_places_048 | verify that the `component.stop` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 49 | NegSem_160104_invoking_functions_from_specific_places_049 | verify that the `kill` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 50 | NegSem_160104_invoking_functions_from_specific_places_050 | verify that the `component.running` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 51 | NegSem_160104_invoking_functions_from_specific_places_051 | verify that the `alive` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 52 | NegSem_160104_invoking_functions_from_specific_places_052 | verify that the `done` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 53 | NegSem_160104_invoking_functions_from_specific_places_053 | verify that the `killed` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 54 | NegSem_160104_invoking_functions_from_specific_places_054 | verify that the `port.start` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 55 | NegSem_160104_invoking_functions_from_specific_places_055 | verify that the `port.stop` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 56 | NegSem_160104_invoking_functions_from_specific_places_056 | verify that the `halt` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 57 | NegSem_160104_invoking_functions_from_specific_places_057 | verify that the `clear` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 58 | NegSem_160104_invoking_functions_from_specific_places_058 | verify that the `checkstate` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 59 | NegSem_160104_invoking_functions_from_specific_places_059 | verify that the `send` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 60 | NegSem_160104_invoking_functions_from_specific_places_060 | verify that the `receive` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 61 | NegSem_160104_invoking_functions_from_specific_places_061 | verify that the `trigger` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 62 | NegSem_160104_invoking_functions_from_specific_places_062 | verify that the `call` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 63 | NegSem_160104_invoking_functions_from_specific_places_063 | verify that the `getcall` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 64 | NegSem_160104_invoking_functions_from_specific_places_064 | verify that the `reply` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 65 | NegSem_160104_invoking_functions_from_specific_places_065 | verify that the `getreply` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 66 | NegSem_160104_invoking_functions_from_specific_places_066 | verify that the `raise` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 67 | NegSem_160104_invoking_functions_from_specific_places_067 | verify that the `catch` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 68 | NegSem_160104_invoking_functions_from_specific_places_068 | verify that the `check` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 69 | NegSem_160104_invoking_functions_from_specific_places_069 | verify that the `connect` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 70 | NegSem_160104_invoking_functions_from_specific_places_070 | verify that the `disconnect` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 71 | NegSem_160104_invoking_functions_from_specific_places_071 | verify that the `map` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 72 | NegSem_160104_invoking_functions_from_specific_places_072 | verify that the `unmap` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 73 | NegSem_160104_invoking_functions_from_specific_places_073 | verify that the `action` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 74 | NegSem_160104_invoking_functions_from_specific_places_074 | verify that the `timer.start` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 75 | NegSem_160104_invoking_functions_from_specific_places_075 | verify that the `timer.stop` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 76 | NegSem_160104_invoking_functions_from_specific_places_076 | verify that the `timer.running` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 77 | NegSem_160104_invoking_functions_from_specific_places_077 | verify that the `read` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 78 | NegSem_160104_invoking_functions_from_specific_places_078 | verify that the `timeout` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 79 | NegSem_160104_invoking_functions_from_specific_places_079 | verify that a non-deterministic external function call cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 80 | NegSem_160104_invoking_functions_from_specific_places_080 | verify that the predefined `rnd` function cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 81 | NegSem_160104_invoking_functions_from_specific_places_081 | verify a function called during receiving operation cannot contain an assignment of a component variable (in template fields) | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 82 | NegSem_160104_invoking_functions_from_specific_places_082 | verify a function called during receiving operation cannot contain a component variable used as an actual `out` parameter (in template fields) | Clause 16.1.4 | m | n |
| 83 | NegSem_160104_invoking_functions_from_specific_places_083 | verify a function called during receiving operation cannot contain a component variable used as an actual `inout` parameter (in template fields) | Clause 16.1.4 | m | n |
| 84 | NegSem_160104_invoking_functions_from_specific_places_084 | verify that the `setverdict` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 85 | NegSem_160104_invoking_functions_from_specific_places_085 | verify that the `activate` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 86 | NegSem_160104_invoking_functions_from_specific_places_086 | verify that the `deactivate` operation cannot be used in a function called during receiving operation (in template fields) | Clause 16.1.4 | m | n |
| 87 | NegSem_160104_invoking_functions_from_specific_places_087 | verify that the `create` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 88 | NegSem_160104_invoking_functions_from_specific_places_088 | verify that the `component.start` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 89 | NegSem_160104_invoking_functions_from_specific_places_089 | verify that the `component.stop` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 90 | NegSem_160104_invoking_functions_from_specific_places_090 | verify that the `kill` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 91 | NegSem_160104_invoking_functions_from_specific_places_091 | verify that the `component.running` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 92 | NegSem_160104_invoking_functions_from_specific_places_092 | verify that the `alive` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 93 | NegSem_160104_invoking_functions_from_specific_places_093 | verify that the `done` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 94 | NegSem_160104_invoking_functions_from_specific_places_094 | verify that the `killed` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 95 | NegSem_160104_invoking_functions_from_specific_places_095 | verify that the `port.start` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 96 | NegSem_160104_invoking_functions_from_specific_places_096 | verify that the `port.stop` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 97 | NegSem_160104_invoking_functions_from_specific_places_097 | verify that the `halt` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 98 | NegSem_160104_invoking_functions_from_specific_places_098 | verify that the `clear` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 99 | NegSem_160104_invoking_functions_from_specific_places_099 | verify that the `checkstate` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 100 | NegSem_160104_invoking_functions_from_specific_places_100 | verify that the `send` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 101 | NegSem_160104_invoking_functions_from_specific_places_101 | verify that the `receive` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 102 | NegSem_160104_invoking_functions_from_specific_places_102 | verify that the `trigger` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 103 | NegSem_160104_invoking_functions_from_specific_places_103 | verify that the `call` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 104 | NegSem_160104_invoking_functions_from_specific_places_104 | verify that the `getcall` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 105 | NegSem_160104_invoking_functions_from_specific_places_105 | verify that the `reply` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 106 | NegSem_160104_invoking_functions_from_specific_places_106 | verify that the `getreply` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 107 | NegSem_160104_invoking_functions_from_specific_places_107 | verify that the `raise` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 108 | NegSem_160104_invoking_functions_from_specific_places_108 | verify that the `catch` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 109 | NegSem_160104_invoking_functions_from_specific_places_109 | verify that the `check` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 110 | NegSem_160104_invoking_functions_from_specific_places_110 | verify that the `connect` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 111 | NegSem_160104_invoking_functions_from_specific_places_111 | verify that the `disconnect` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 112 | NegSem_160104_invoking_functions_from_specific_places_112 | verify that the `map` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 113 | NegSem_160104_invoking_functions_from_specific_places_113 | verify that the `unmap` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 114 | NegSem_160104_invoking_functions_from_specific_places_114 | verify that the `action` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 115 | NegSem_160104_invoking_functions_from_specific_places_115 | verify that the `timer.start` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 116 | NegSem_160104_invoking_functions_from_specific_places_116 | verify that the `timer.stop` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 117 | NegSem_160104_invoking_functions_from_specific_places_117 | verify that the `timer.running` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 118 | NegSem_160104_invoking_functions_from_specific_places_118 | verify that the `read` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 119 | NegSem_160104_invoking_functions_from_specific_places_119 | verify that the `timeout` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 120 | NegSem_160104_invoking_functions_from_specific_places_120 | verify that a non-deterministic external function call cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 121 | NegSem_160104_invoking_functions_from_specific_places_121 | verify that the predefined `rnd` function cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 122 | NegSem_160104_invoking_functions_from_specific_places_122 | verify a function called during receiving operation cannot contain an assignment of a component variable (in in-line templates) | Clause 16.1.4 | m | y |
| 123 | NegSem_160104_invoking_functions_from_specific_places_123 | verify a function called during receiving operation cannot contain a component variable used as an actual `out` parameter (in in-line templates) | Clause 16.1.4 | m | y |
| 124 | NegSem_160104_invoking_functions_from_specific_places_124 | verify a function called during receiving operation cannot contain a component variable used as an actual `inout` parameter (in in-line templates) | Clause 16.1.4 | m | y |
| 125 | NegSem_160104_invoking_functions_from_specific_places_125 | verify that the `setverdict` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 126 | NegSem_160104_invoking_functions_from_specific_places_126 | verify that the `activate` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 127 | NegSem_160104_invoking_functions_from_specific_places_127 | verify that the `deactivate` operation cannot be used in a function called during receiving operation (in in-line templates) | Clause 16.1.4 | m | y |
| 128 | NegSem_160104_invoking_functions_from_specific_places_128 | verify a function called during receiving operation cannot contain an `out` parameter (in in-line templates) | Clause 16.1.4 | m | y |
| 129 | NegSem_160104_invoking_functions_from_specific_places_129 | verify a function called during receiving operation cannot contain an `inout` parameter (in in-line templates) | Clause 16.1.4 | m | y |
| 130 | NegSem_160104_invoking_functions_from_specific_places_130 | verify that the `create` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 131 | NegSem_160104_invoking_functions_from_specific_places_131 | verify that the `component.start` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 132 | NegSem_160104_invoking_functions_from_specific_places_132 | verify that the `component.stop` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 133 | NegSem_160104_invoking_functions_from_specific_places_133 | verify that the `kill` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 134 | NegSem_160104_invoking_functions_from_specific_places_134 | verify that the `component.running` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 135 | NegSem_160104_invoking_functions_from_specific_places_135 | verify that the `alive` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 136 | NegSem_160104_invoking_functions_from_specific_places_136 | verify that the `done` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 137 | NegSem_160104_invoking_functions_from_specific_places_137 | verify that the `killed` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 138 | NegSem_160104_invoking_functions_from_specific_places_138 | verify that the `port.start` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 139 | NegSem_160104_invoking_functions_from_specific_places_139 | verify that the `port.stop` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 140 | NegSem_160104_invoking_functions_from_specific_places_140 | verify that the `halt` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 141 | NegSem_160104_invoking_functions_from_specific_places_141 | verify that the `clear` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 142 | NegSem_160104_invoking_functions_from_specific_places_142 | verify that the `checkstate` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 143 | NegSem_160104_invoking_functions_from_specific_places_143 | verify that the `send` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 144 | NegSem_160104_invoking_functions_from_specific_places_144 | verify that the `receive` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 145 | NegSem_160104_invoking_functions_from_specific_places_145 | verify that the `trigger` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 146 | NegSem_160104_invoking_functions_from_specific_places_146 | verify that the `call` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 147 | NegSem_160104_invoking_functions_from_specific_places_147 | verify that the `getcall` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 148 | NegSem_160104_invoking_functions_from_specific_places_148 | verify that the `reply` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 149 | NegSem_160104_invoking_functions_from_specific_places_149 | verify that the `getreply` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 150 | NegSem_160104_invoking_functions_from_specific_places_150 | verify that the `raise` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 151 | NegSem_160104_invoking_functions_from_specific_places_151 | verify that the `catch` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 152 | NegSem_160104_invoking_functions_from_specific_places_152 | verify that the `check` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 153 | NegSem_160104_invoking_functions_from_specific_places_153 | verify that the `connect` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 154 | NegSem_160104_invoking_functions_from_specific_places_154 | verify that the `disconnect` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 155 | NegSem_160104_invoking_functions_from_specific_places_155 | verify that the `map` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 156 | NegSem_160104_invoking_functions_from_specific_places_156 | verify that the `unmap` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 157 | NegSem_160104_invoking_functions_from_specific_places_157 | verify that the `action` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 158 | NegSem_160104_invoking_functions_from_specific_places_158 | verify that the `timer.start` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 159 | NegSem_160104_invoking_functions_from_specific_places_159 | verify that the `timer.stop` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 160 | NegSem_160104_invoking_functions_from_specific_places_160 | verify that the `timer.running` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 161 | NegSem_160104_invoking_functions_from_specific_places_161 | verify that the `read` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 162 | NegSem_160104_invoking_functions_from_specific_places_162 | verify that the `timeout` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 163 | NegSem_160104_invoking_functions_from_specific_places_163 | verify that a non-deterministic external function call cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 164 | NegSem_160104_invoking_functions_from_specific_places_164 | verify that the predefined `rnd` function cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 165 | NegSem_160104_invoking_functions_from_specific_places_165 | verify a function called during receiving operation cannot contain an assignment of a component variable (as actual parameters) | Clause 16.1.4 | m | y |
| 166 | NegSem_160104_invoking_functions_from_specific_places_166 | verify a function called during receiving operation cannot contain a component variable used as an actual out parameter (as actual parameters) | Clause 16.1.4 | m | y |
| 167 | NegSem_160104_invoking_functions_from_specific_places_167 | verify a function called during receiving operation cannot contain a component variable used as an actual `inout` parameter (as actual parameters) | Clause 16.1.4 | m | y |
| 168 | NegSem_160104_invoking_functions_from_specific_places_168 | verify that the `setverdict` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 169 | NegSem_160104_invoking_functions_from_specific_places_169 | verify that the `activate` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 170 | NegSem_160104_invoking_functions_from_specific_places_170 | verify that the `deactivate` operation cannot be used in a function called during receiving operation (as actual parameters) | Clause 16.1.4 | m | y |
| 171 | NegSem_160104_invoking_functions_from_specific_places_171 | verify a function called during receiving operation cannot contain an out parameter (as actual parameters) | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 172 | NegSem_160104_invoking_functions_from_specific_places_172 | verify a function called during receiving operation cannot contain an `inout` parameter (as actual parameters) | Clause 16.1.4 | m | y |
| 173 | NegSem_160104_invoking_functions_from_specific_places_173 | verify that the `create` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 174 | NegSem_160104_invoking_functions_from_specific_places_174 | verify that the `component.start` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 175 | NegSem_160104_invoking_functions_from_specific_places_175 | verify that the `component.stop` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 176 | NegSem_160104_invoking_functions_from_specific_places_176 | verify that the `kill` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 177 | NegSem_160104_invoking_functions_from_specific_places_177 | verify that the `component.running` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 178 | NegSem_160104_invoking_functions_from_specific_places_178 | verify that the `alive` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 179 | NegSem_160104_invoking_functions_from_specific_places_179 | verify that the `done` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 180 | NegSem_160104_invoking_functions_from_specific_places_180 | verify that the `killed` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 181 | NegSem_160104_invoking_functions_from_specific_places_181 | verify that the `port.start` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 182 | NegSem_160104_invoking_functions_from_specific_places_182 | verify that the `port.stop` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 183 | NegSem_160104_invoking_functions_from_specific_places_183 | verify that the `halt` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 184 | NegSem_160104_invoking_functions_from_specific_places_184 | verify that the `clear` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 185 | NegSem_160104_invoking_functions_from_specific_places_185 | verify that the `checkstate` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 186 | NegSem_160104_invoking_functions_from_specific_places_186 | verify that the `send` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 187 | NegSem_160104_invoking_functions_from_specific_places_187 | verify that the `receive` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 188 | NegSem_160104_invoking_functions_from_specific_places_188 | verify that the `trigger` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 189 | NegSem_160104_invoking_functions_from_specific_places_189 | verify that the `call` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 190 | NegSem_160104_invoking_functions_from_specific_places_190 | verify that the `getcall` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 191 | NegSem_160104_invoking_functions_from_specific_places_191 | verify that the `reply` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 192 | NegSem_160104_invoking_functions_from_specific_places_192 | verify that the `getreply` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 193 | NegSem_160104_invoking_functions_from_specific_places_193 | verify that the `raise` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 194 | NegSem_160104_invoking_functions_from_specific_places_194 | verify that the `catch` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 195 | NegSem_160104_invoking_functions_from_specific_places_195 | verify that the `check` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 196 | NegSem_160104_invoking_functions_from_specific_places_196 | verify that the `connect` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 197 | NegSem_160104_invoking_functions_from_specific_places_197 | verify that the `disconnect` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 198 | NegSem_160104_invoking_functions_from_specific_places_198 | verify that the `map` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 199 | NegSem_160104_invoking_functions_from_specific_places_199 | verify that the `unmap` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 200 | NegSem_160104_invoking_functions_from_specific_places_200 | verify that the `action` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 201 | NegSem_160104_invoking_functions_from_specific_places_201 | verify that the `timer.start` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 202 | NegSem_160104_invoking_functions_from_specific_places_202 | verify that the `timer.stop` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 203 | NegSem_160104_invoking_functions_from_specific_places_203 | verify that the `timer.running` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 204 | NegSem_160104_invoking_functions_from_specific_places_204 | verify that the `read` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 205 | NegSem_160104_invoking_functions_from_specific_places_205 | verify that the `timeout` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 206 | NegSem_160104_invoking_functions_from_specific_places_206 | verify a function called in a guard of an `alt` statement cannot contain an assignment of a component variable | Clause 16.1.4 | m | y |
| 207 | NegSem_160104_invoking_functions_from_specific_places_207 | verify a function called in a guard of an `alt` statement cannot contain a component variable used as an actual `out` parameter | Clause 16.1.4 | m | y |
| 208 | NegSem_160104_invoking_functions_from_specific_places_208 | verify a function called in a guard of an `alt` statement cannot contain a component variable used as an actual `inout` parameter | Clause 16.1.4 | m | y |
| 209 | NegSem_160104_invoking_functions_from_specific_places_209 | verify that the `activate` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 210 | NegSem_160104_invoking_functions_from_specific_places_210 | verify that the `deactivate` operation cannot be used in guards of `alt` statements | Clause 16.1.4 | m | y |
| 211 | NegSem_160104_invoking_functions_from_specific_places_211 | verify that a function called from a `guard` statement of an `alt` operation cannot contain `out` parameters | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 212 | NegSem_160104_invoking_functions_from_specific_places_212 | verify that the `create` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 213 | NegSem_160104_invoking_functions_from_specific_places_213 | verify that the `component.start` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 214 | NegSem_160104_invoking_functions_from_specific_places_214 | verify that the `component.stop` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 215 | NegSem_160104_invoking_functions_from_specific_places_215 | verify that the `kill` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 216 | NegSem_160104_invoking_functions_from_specific_places_216 | verify that the `component.running` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 217 | NegSem_160104_invoking_functions_from_specific_places_217 | verify that the `alive` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 218 | NegSem_160104_invoking_functions_from_specific_places_218 | verify that the `done` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 219 | NegSem_160104_invoking_functions_from_specific_places_219 | verify that the `killed` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 220 | NegSem_160104_invoking_functions_from_specific_places_220 | verify that the `port.start` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 221 | NegSem_160104_invoking_functions_from_specific_places_221 | verify that the `port.stop` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 222 | NegSem_160104_invoking_functions_from_specific_places_222 | verify that the `halt` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 223 | NegSem_160104_invoking_functions_from_specific_places_223 | verify that the `clear` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 224 | NegSem_160104_invoking_functions_from_specific_places_224 | verify that the `checkstate` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 225 | NegSem_160104_invoking_functions_from_specific_places_225 | verify that the `send` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 226 | NegSem_160104_invoking_functions_from_specific_places_226 | verify that the `receive` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 227 | NegSem_160104_invoking_functions_from_specific_places_227 | verify that the `trigger` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 228 | NegSem_160104_invoking_functions_from_specific_places_228 | verify that the `call` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 229 | NegSem_160104_invoking_functions_from_specific_places_229 | verify that the `getcall` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 230 | NegSem_160104_invoking_functions_from_specific_places_230 | verify that the `reply` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 231 | NegSem_160104_invoking_functions_from_specific_places_231 | verify that the `getreply` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 232 | NegSem_160104_invoking_functions_from_specific_places_232 | verify that the `raise` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 233 | NegSem_160104_invoking_functions_from_specific_places_233 | verify that the `catch` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 234 | NegSem_160104_invoking_functions_from_specific_places_234 | verify that the `check` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 235 | NegSem_160104_invoking_functions_from_specific_places_235 | verify that the `connect` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 236 | NegSem_160104_invoking_functions_from_specific_places_236 | verify that the `disconnect` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 237 | NegSem_160104_invoking_functions_from_specific_places_237 | verify that the `map` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 238 | NegSem_160104_invoking_functions_from_specific_places_238 | verify that the `unmap` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 239 | NegSem_160104_invoking_functions_from_specific_places_239 | verify that the `action` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 240 | NegSem_160104_invoking_functions_from_specific_places_240 | verify that the `timer.start` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 241 | NegSem_160104_invoking_functions_from_specific_places_241 | verify that the `timer.stop` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 242 | NegSem_160104_invoking_functions_from_specific_places_242 | verify that the `timer.running` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 243 | NegSem_160104_invoking_functions_from_specific_places_243 | verify that the `read` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 244 | NegSem_160104_invoking_functions_from_specific_places_244 | verify that the `timeout` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 245 | NegSem_160104_invoking_functions_from_specific_places_245 | verify that a non-deterministic external function call cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 246 | NegSem_160104_invoking_functions_from_specific_places_246 | verify that the predefined `rnd` function cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 247 | NegSem_160104_invoking_functions_from_specific_places_247 | verify a function called in a guard of an `altstep` cannot contain an assignment of a component variable | Clause 16.1.4 | m | y |
| 248 | NegSem_160104_invoking_functions_from_specific_places_248 | verify a function called in a guard of an `altstep` cannot contain a component variable used as an actual `out` parameter | Clause 16.1.4 | m | y |
| 249 | NegSem_160104_invoking_functions_from_specific_places_249 | verify a function called in a guard of an `altstep` cannot contain a component variable used as an actual `inout` parameter | Clause 16.1.4 | m | y |
| 250 | NegSem_160104_invoking_functions_from_specific_places_250 | verify that the `setverdict` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 251 | NegSem_160104_invoking_functions_from_specific_places_251 | verify that the `activate` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 252 | NegSem_160104_invoking_functions_from_specific_places_252 | verify that the `deactivate` operation cannot be used in guards of altsteps | Clause 16.1.4 | m | y |
| 253 | NegSem_160104_invoking_functions_from_specific_places_253 | verify that a function called from a `guard` statement of an altstep cannot contain `out` parameters | Clause 16.1.4 | m | y |
| 254 | NegSem_160104_invoking_functions_from_specific_places_254 | verify that a function called from a `guard` statement of an altstep cannot contain `inout` parameters | Clause 16.1.4 | m | y |
| 255 | NegSem_160104_invoking_functions_from_specific_places_255 | verify that the `create` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 256 | NegSem_160104_invoking_functions_from_specific_places_256 | verify that the `component.start` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 257 | NegSem_160104_invoking_functions_from_specific_places_257 | verify that the `component.stop` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 258 | NegSem_160104_invoking_functions_from_specific_places_258 | verify that the `kill` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 259 | NegSem_160104_invoking_functions_from_specific_places_259 | verify that the `component.running` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 260 | NegSem_160104_invoking_functions_from_specific_places_260 | verify that the `alive` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 261 | NegSem_160104_invoking_functions_from_specific_places_261 | verify that the `done` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 262 | NegSem_160104_invoking_functions_from_specific_places_262 | verify that the `killed` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 263 | NegSem_160104_invoking_functions_from_specific_places_263 | verify that the `port.start` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 264 | NegSem_160104_invoking_functions_from_specific_places_264 | verify that the `port.stop` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 265 | NegSem_160104_invoking_functions_from_specific_places_265 | verify that the `halt` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 266 | NegSem_160104_invoking_functions_from_specific_places_266 | verify that the `clear` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 267 | NegSem_160104_invoking_functions_from_specific_places_267 | verify that the `checkstate` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 268 | NegSem_160104_invoking_functions_from_specific_places_268 | verify that the `send` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 269 | NegSem_160104_invoking_functions_from_specific_places_269 | verify that the `receive` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 270 | NegSem_160104_invoking_functions_from_specific_places_270 | verify that the `trigger` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 271 | NegSem_160104_invoking_functions_from_specific_places_271 | verify that the `call` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 272 | NegSem_160104_invoking_functions_from_specific_places_272 | verify that the `getcall` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 273 | NegSem_160104_invoking_functions_from_specific_places_273 | verify that the `reply` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 274 | NegSem_160104_invoking_functions_from_specific_places_274 | verify that the `getreply` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 275 | NegSem_160104_invoking_functions_from_specific_places_275 | verify that the `raise` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 276 | NegSem_160104_invoking_functions_from_specific_places_276 | verify that the `catch` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 277 | NegSem_160104_invoking_functions_from_specific_places_277 | verify that the `check` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 278 | NegSem_160104_invoking_functions_from_specific_places_278 | verify that the `connect` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 279 | NegSem_160104_invoking_functions_from_specific_places_279 | verify that the `disconnect` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 280 | NegSem_160104_invoking_functions_from_specific_places_280 | verify that the `map` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 281 | NegSem_160104_invoking_functions_from_specific_places_281 | verify that the `unmap` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 282 | NegSem_160104_invoking_functions_from_specific_places_282 | verify that the `action` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 283 | NegSem_160104_invoking_functions_from_specific_places_283 | verify that the `timer.start` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 284 | NegSem_160104_invoking_functions_from_specific_places_284 | verify that the `timer.stop` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 285 | NegSem_160104_invoking_functions_from_specific_places_285 | verify that the `timer.running` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 286 | NegSem_160104_invoking_functions_from_specific_places_286 | verify that the `read` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 287 | NegSem_160104_invoking_functions_from_specific_places_287 | verify that the `timeout` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 288 | NegSem_160104_invoking_functions_from_specific_places_288 | verify that a non-deterministic external function call cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 289 | NegSem_160104_invoking_functions_from_specific_places_289 | verify that the predefined `rnd` function cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 290 | NegSem_160104_invoking_functions_from_specific_places_290 | verify a function called in an altstep local definition cannot contain an assignment of a component variable | Clause 16.1.4 | m | n |
| 291 | NegSem_160104_invoking_functions_from_specific_places_291 | verify a function called in an altstep local definition cannot contain a component variable used as an actual `out` parameter | Clause 16.1.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 292 | NegSem_160104_invoking_functions_from_specific_places_292 | verify a function called in an altstep local definition cannot contain a component variable used as an actual `inout` parameter | Clause 16.1.4 | m | n |
| 293 | NegSem_160104_invoking_functions_from_specific_places_293 | verify that the `setverdict` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 294 | NegSem_160104_invoking_functions_from_specific_places_294 | verify that the `activate` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 295 | NegSem_160104_invoking_functions_from_specific_places_295 | verify that the `deactivate` operation cannot be used in altstep local definitions | Clause 16.1.4 | m | n |
| 296 | NegSem_160104_invoking_functions_from_specific_places_296 | verify that a function called in altstep `altstep` local definitions cannot contain `out` parameters | Clause 16.1.4 | m | n |
| 297 | NegSem_160104_invoking_functions_from_specific_places_297 | verify that a function called in altstep `altstep` local definitions cannot contain `inout` parameters | Clause 16.1.4 | m | n |

# 5.100. Altsteps

*Table 99. Altsteps*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1602_toplevel_001 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_1602_toplevel_002 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | n |
| 3 | NegSem_1602_toplevel_003 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | n |
| 4 | NegSem_1602_toplevel_004 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | n |
| 5 | NegSem_1602_toplevel_005 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | y |
| 6 | NegSem_1602_toplevel_006 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | y |
| 7 | NegSem_1602_toplevel_007 | Verify that altstep without a `runs on` clause cannot be started as component behaviour | Clause 16.2 | m | n |
| 8 | NegSyn_1602_toplevel_001 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | y |
| 9 | Sem_1602_toplevel_001 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2 | m | y |
| 10 | Sem_1602_toplevel_002 | Verify that altstep with a `runs on` clause can be started as component behaviour | Clause 16.2 | m | n |
| 11 | Sem_1602_toplevel_003 | Verify that altstep with a `runs on` clause can be started as component behaviour from a context without a `runs on` clause | Clause 16.2 | m | n |

# 5.101. Invoking altsteps

*Table 100. Invoking altsteps*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_160201_invoking_altsteps_001 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2.1 | m | y |
| 2 | Sem_160201_invoking_altsteps_001 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2.1 | m | y |
| 3 | Sem_160201_invoking_altsteps_002 | The IUT recognizes altstep definitions and correctly evaluates them | Clause 16.2.1 | m | y |
| 4 | Sem_160201_invoking_altsteps_003 | Altsteps are correctly handled for dynamically mapped ports | Clause 16.2.1 | m | y |
| 5 | Sem_160201_invoking_altsteps_004 | Altsteps are correctly handled for dynamically mapped ports | Clause 16.2.1 | m | y |

## 5.102. Test cases

*Table 101. Test cases*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1603_testcases_001 | The IUT properly evaluates invocation of testcases | Clause 16.3 | m | y |
| 2 | NegSem_1603_testcases_002 | The IUT properly evaluates invocation of testcases | Clause 16.3 | m | y |
| 3 | Syn_1603_testcases_001 | The IUT properly evaluates invocation of testcases with `system` clause | Clause 16.3 | m | y |

## 5.103. Assignments

*Table 102. Assignments*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1901_assignments_001 | The IUT properly evaluates `assignment` statements | Clause 19.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_1901_assignments_002 | The IUT properly evaluates `assignment` statements | Clause 19.1 | m | y |
| 3 | NegSem_1901_assignments_003 | The IUT properly evaluates `assignment` statements | Clause 19.1 | m | y |
| 4 | NegSem_1901_assignments_004 | Omit assignment to a record non-optional value is not allowed | Clause 19.1 | m | y |
| 5 | NegSem_1901_assignments_005 | Omit assignment to set of non-optional value is not allowed | Clause 19.1 | m | y |
| 6 | NegSem_1901_assignments_006 | Omit assignment to an array is not allowed | Clause 19.1 | m | y |
| 7 | NegSyn_1901_assignments_001 | The IUT properly evaluates `assignment` statements | Clause 19.1 | m | y |
| 8 | Sem_1901_assignments_001 | The IUT properly evaluates `assignment` statements | Clause 19.1 | m | y |
| 9 | Sem_1901_assignments_002 | Uninitialized at the right-hand side of the assignment shall also become uninitialized at the left-hand side | Clause 19.1 | m | y |
| 10 | Sem_1901_assignments_003 | The right-hand side of the assignment of a structured value is evaulted correctly | Clause 19.1 | m | y |
| 11 | Sem_1901_assignments_004 | Ensure that the right-hand side of the assignment of a structured value is evaulted correctly | Clause 19.1 | m | n |

## 5.104. The `if-else` statement

*Table 103. The `if-else` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_1902_if_else_statement_001 | If statement requires curly brackets for the body | Clause 19.2 | m | y |
| 2 | Sem_1902_if_else_statement_001 | The IUT properly evaluates `if-else` statements | Clause 19.2 | m | y |
| 3 | Sem_1902_if_else_statement_002 | The IUT properly evaluates `if-else` statements | Clause 19.2 | m | y |

## 5.105. The `Select` statements

*Table 104. The `Select` statements*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_190301_select_case_statement_001 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |
| 2 | Sem_190301_select_case_statement_002 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |
| 3 | Sem_190301_select_case_statement_003 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |
| 4 | Sem_190301_select_case_statement_004 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |
| 5 | Sem_190301_select_case_statement_005 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |
| 6 | Sem_190301_select_case_statement_006 | The IUT properly evaluates `select-case` statements | Clause 19.3 | m | y |

## 5.106. The `select union` statement

*Table 105. The `select union` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_190302_select_union_statement_001 | verify that header part of `select-union` statements cannot contain anything else than union instances | Clause 19.3.2 | m | y |
| 2 | NegSem_190302_select_union_statement_002 | verify that uninitialized value cannot be used in select union header | Clause 19.3.2 | m | y |
| 3 | NegSem_190302_select_union_statement_003 | verify that unknown alternatives cannot be use in `case` statements | Clause 19.3.2 | m | y |
| 4 | NegSem_190302_select_union_statement_004 | verify that the same alternative cannot be used in two `case` statements (simple case) | Clause 19.3.2 | m | y |
| 5 | NegSem_190302_select_union_statement_005 | verify that the same alternative cannot be used in two `case` statements (list item) | Clause 19.3.2 | m | y |
| 6 | NegSem_190302_select_union_statement_006 | verify that it is possible to use a `select union` statement with several branches | Clause 19.3.2 | m | y |
| 7 | Sem_190302_select_union_statement_001 | verify that it is possible to use a `select union` statement with several branches | Clause 19.3.2 | m | y |
| 8 | Sem_190302_select_union_statement_002 | verify that it is possible to use comma separated list of alternatives in case branches | Clause 19.3.2 | m | y |
| 9 | Sem_190302_select_union_statement_003 | verify that it is possible to use an else branches | Clause 19.3.2 | m | y |
| 10 | Sem_190302_select_union_statement_004 | verify that else branch is executed if no case is defined for the selected alternative | Clause 19.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_190302_select_union_statement_005 | verify that no branch is executed if there's no suitable case branch | Clause 19.3.2 | m | y |
| 12 | Sem_190302_select_union_statement_006 | verify that partially initialized value can be used in select union header | Clause 19.3.2 | m | y |
| 13 | Sem_190302_select_union_statement_007 | verify that it is possible to use a `select union` statement with several branches | Clause 19.3.2 | m | y |

## 5.107. The `for` statement

*Table 106. The `for` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1904_for_statement_001 | The IUT properly evaluates `for` statements | Clause 19.4 | m | y |
| 2 | Sem_1904_for_statement_001 | The IUT properly evaluates `for` statements | Clause 19.4 | m | y |
| 3 | Sem_1904_for_statement_002 | The IUT properly evaluates `for` statements | Clause 19.4 | m | y |
| 4 | Sem_1904_for_statement_003 | The IUT properly evaluates `for` statements | Clause 19.4 | m | y |

## 5.108. The `while` statement

*Table 107. The `while` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1905_while_statement_001 | The IUT properly evaluates `while` statements | Clause 19.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_1905_while_statement_001 | The IUT properly evaluates `while` statements | Clause 19.5 | m | y |
| 3 | Sem_1905_while_statement_002 | The IUT properly evaluates `while` statements | Clause 19.5 | m | y |
| 4 | Sem_1905_while_statement_003 | The IUT properly evaluates `while` statements | Clause 19.5 | m | y |

## 5.109. The `do-while` statement

*Table 108. The `do-while` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1906_do_while_statement_001 | The IUT properly evaluates `do-while` statements | Clause 19.6 | m | y |
| 2 | Sem_1906_do_while_statement_001 | The IUT properly evaluates `do-while` statements | Clause 19.6 | m | y |
| 3 | Sem_1906_do_while_statement_002 | The IUT properly evaluates `do-while` statements | Clause 19.6 | m | y |
| 4 | Sem_1906_do_while_statement_003 | The IUT properly evaluates `do-while` statements | Clause 19.6 | m | y |

## 5.110. The `label` statement

*Table 109. The `label` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1907_label_statement_001 | The IUT correctly handles label naming uniqueness. | Clause 19.7 | m | y |
| 2 | NegSyn_1907_label_statement_001 | The IUT correctly handles label syntax. | Clause 19.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSyn_1907_label_statement_002 | The IUT correctly handles label syntax. | Clause 19.7 | m | y |
| 4 | Syn_1907_label_statement_001 | The IUT correctly handles label syntax. | Clause 19.7 | m | y |

## 5.111. The `goto` statement

*Table 110. The `goto` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1908_goto_statement_001 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |
| 2 | NegSem_1908_goto_statement_002 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |
| 3 | NegSem_1908_goto_statement_003 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |
| 4 | Sem_1908_goto_statement_001 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |
| 5 | Sem_1908_goto_statement_002 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |
| 6 | Sem_1908_goto_statement_003 | The IUT correctly handles `goto` statements. | Clause 19.8 | m | y |

## 5.112. The `stop` execution statement

*Table 111. The `stop` execution statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_1909_stop_statement_001 | The IUT correctly handles `stop` statements. | Clause 19.9 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_1909_stop_statement_002 | The IUT correctly handles `stop` statements. | Clause 19.9 | m | y |
| 3 | Sem_1909_stop_statement_003 | `stop` statement in a function called from a PTC | Clause 19.9 | m | y |
| 4 | Sem_1909_stop_statement_004 | `stop` statement in a function called from a PTC | Clause 19.9 | m | y |

## 5.113. The `return` statement

*Table 112. The `return` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1910_return_statement_001 | The IUT correctly handles `return` statements. | Clause 19.10 | m | y |
| 2 | Sem_1910_return_statement_001 | The IUT correctly handles `return` statements. | Clause 19.10 | m | y |
| 3 | Sem_1910_return_statement_002 | The IUT correctly handles `return` statements. | Clause 19.10 | m | y |

## 5.114. The `log` statement

*Table 113. The `log` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_1911_log_statement_001 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |
| 2 | Sem_1911_log_statement_001 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |
| 3 | Sem_1911_log_statement_002 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_1911_log_statement_003 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |
| 5 | Sem_1911_log_statement_004 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |
| 6 | Sem_1911_log_statement_005 | The IUT properly evaluates `log` statements | Clause 19.11 | m | y |

## 5.115. The `continue` statement

*Table 114. The `continue` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_1913_continue_statement_001 | The IUT properly evaluates `continue` statements | Clause 19.13 | m | y |

## 5.116. Statement and operations for alternative behaviours

*Table 115. Statement and operations for alternative behaviours*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_20_TopLevel_001 | `Alt`-statements are accepted. | Clause 20 | m | y |
| 2 | Syn_20_TopLevel_002 | Repeat in an `alt` -statement is accepted. | Clause 20 | m | y |
| 3 | Syn_20_TopLevel_003 | The `interleave`-statement is accepted. | Clause 20 | m | y |
| 4 | Syn_20_TopLevel_004 | Defaults and the `activate` statement is accepted. | Clause 20 | m | y |
| 5 | Syn_20_TopLevel_005 | Defaults and the `activate` statement is accepted. | Clause 20 | m | y |

## 5.117. The `alt` statement

*Table 116. The `alt` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2002_TheAltStatement_001 | dynamic error if a test component is completely blocked | Clause 20.2 | m | y |
| 2 | NegSem_2002_TheAltStatement_002 | create in `guard` statements | Clause 20.2 | m | n |
| 3 | NegSem_2002_TheAltStatement_003 | running (timer) in `guard` statements | Clause 20.2 | m | n |
| 4 | NegSem_2002_TheAltStatement_004 | running (component) in `guard` statements | Clause 20.2 | m | n |
| 5 | NegSem_2002_TheAltStatement_005 | alive in `guard` statements | Clause 20.2 | m | n |
| 6 | NegSem_2002_TheAltStatement_006 | activate in `guard` statements | Clause 20.2 | m | n |
| 7 | NegSem_2002_TheAltStatement_007 | create in `alt` branch event | Clause 20.2 | m | n |
| 8 | NegSem_2002_TheAltStatement_008 | running (timer) in `alt` branch event | Clause 20.2 | m | n |
| 9 | NegSem_2002_TheAltStatement_009 | running (component) in `alt` branch event | Clause 20.2 | m | n |
| 10 | NegSem_2002_TheAltStatement_010 | alive in `alt` branch event | Clause 20.2 | m | n |
| 11 | NegSem_2002_TheAltStatement_011 | create in `alt` branch event | Clause 20.2 | m | n |
| 12 | NegSem_2002_TheAltStatement_012 | create in altstep branch | Clause 20.2 | m | n |
| 13 | NegSem_2002_TheAltStatement_013 | running (timer) in altstep branch | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | NegSem_2002_TheAltStatement_014 | running (component) in altstep branch | Clause 20.2 | m | n |
| 15 | NegSem_2002_TheAltStatement_015 | alive in altstep branch | Clause 20.2 | m | n |
| 16 | NegSem_2002_TheAltStatement_016 | create in altstep branch | Clause 20.2 | m | n |
| 17 | NegSem_2002_TheAltStatement_017 | verify that the `create` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 18 | NegSem_2002_TheAltStatement_018 | verify that the `component.start` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 19 | NegSem_2002_TheAltStatement_019 | verify that the `component.stop` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 20 | NegSem_2002_TheAltStatement_020 | verify that the `kill` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 21 | NegSem_2002_TheAltStatement_021 | verify that the `component.running` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 22 | NegSem_2002_TheAltStatement_022 | verify that the `alive` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | NegSem_2002_TheAltStatement_023 | verify that the `done` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 24 | NegSem_2002_TheAltStatement_024 | verify that the `killed` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 25 | NegSem_2002_TheAltStatement_025 | verify that the `port.start` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 26 | NegSem_2002_TheAltStatement_026 | verify that the `port.stop` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 27 | NegSem_2002_TheAltStatement_027 | verify that the `halt` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 28 | NegSem_2002_TheAltStatement_028 | verify that the `clear` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 29 | NegSem_2002_TheAltStatement_029 | verify that the `checkstate` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 30 | NegSem_2002_TheAltStatement_030 | verify that the `send` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 31 | NegSem_2002_TheAltStatement_031 | verify that the `receive` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 32 | NegSem_2002_TheAltStatement_032 | verify that the `trigger` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 33 | NegSem_2002_TheAltStatement_033 | verify that the `call` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 34 | NegSem_2002_TheAltStatement_034 | verify that the `getcall` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 35 | NegSem_2002_TheAltStatement_035 | verify that the `reply` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 36 | NegSem_2002_TheAltStatement_036 | verify that the `getreply` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 37 | NegSem_2002_TheAltStatement_037 | verify that the `raise` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 38 | NegSem_2002_TheAltStatement_038 | verify that the `catch` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 39 | NegSem_2002_TheAltStatement_039 | verify that the `check` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 40 | NegSem_2002_TheAltStatement_040 | verify that the `connect` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 41 | NegSem_2002_TheAltStatement_041 | verify that the `disconnect` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 42 | NegSem_2002_TheAltStatement_042 | verify that the `map` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 43 | NegSem_2002_TheAltStatement_043 | verify that the `unmap` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 44 | NegSem_2002_TheAltStatement_044 | verify that the `action` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 45 | NegSem_2002_TheAltStatement_045 | verify that the `timer.start` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 46 | NegSem_2002_TheAltStatement_046 | verify that the `timer.stop` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 47 | NegSem_2002_TheAltStatement_047 | verify that the `timer.running` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 48 | NegSem_2002_TheAltStatement_048 | verify that the `read` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 49 | NegSem_2002_TheAltStatement_049 | verify that the `timeout` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 50 | NegSem_2002_TheAltStatement_050 | verify that a non-deterministic external function call cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 51 | NegSem_2002_TheAltStatement_051 | verify that the predefined `rnd` function cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 52 | NegSem_2002_TheAltStatement_052 | verify a function called in a guard of an `altstep` cannot contain an assignment of a component variable | Clause 20.2 | m | n |
| 53 | NegSem_2002_TheAltStatement_053 | verify a function called in a guard of an `altstep` cannot contain a component variable used as an actual `out` parameter | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 54 | NegSem_2002_TheAltStatement_054 | verify a function called in a guard of an `altstep` cannot contain a component variable used as an actual `inout` parameter | Clause 20.2 | m | n |
| 55 | NegSem_2002_TheAltStatement_055 | verify that the `setverdict` operation cannot be used in `guard` statements of altstep | Clause 20.2 | m | n |
| 56 | NegSem_2002_TheAltStatement_056 | verify that the `activate` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 57 | NegSem_2002_TheAltStatement_057 | verify that the `deactivate` operation cannot be used in parameters of altsteps invoked from an `alt` branch | Clause 20.2 | m | n |
| 58 | NegSem_2002_TheAltStatement_058 | verify that a function used in a parameter of an altstep invoked from an `alt` branch cannot contain out parameters | Clause 20.2 | m | n |
| 59 | NegSem_2002_TheAltStatement_059 | verify that a function used in a parameter of an altstep invoked from an `alt` branch cannot contain `inout` parameters | Clause 20.2 | m | n |
| 60 | NegSem_2002_TheAltStatement_060 | verify that the `read` operation cannot be used in `guard` statements | Clause 20.2 | m | n |
| 61 | NegSem_2002_TheAltStatement_061 | verify that the `checkstate` operation cannot be used in `guard` statements | Clause 20.2 | m | n |
| 62 | NegSem_2002_TheAltStatement_062 | verify that the `read` operation cannot be used in `alt` branch events (in inline templates) | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 63 | NegSem_2002_TheAltStatement_063 | verify that the `checkstate` operation cannot be used in `alt` branch events (in inline templates) | Clause 20.2 | m | n |
| 64 | NegSem_2002_TheAltStatement_064 | verify that the `read` operation cannot be used in parameters of `alt` branch events | Clause 20.2 | m | n |
| 65 | NegSem_2002_TheAltStatement_065 | verify that the `checkstate` operation cannot be used in parameters of `alt` branch events | Clause 20.2 | m | n |
| 66 | NegSem_2002_TheAltStatement_066 | verify that the `create` operation cannot be used in `alt` branch events (in inline template) | Clause 20.2 | m | n |
| 67 | NegSem_2002_TheAltStatement_067 | verify that the `component.running` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |
| 68 | NegSem_2002_TheAltStatement_068 | verify that the `alive` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |
| 69 | NegSem_2002_TheAltStatement_069 | verify that the `checkstate` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |
| 70 | NegSem_2002_TheAltStatement_070 | verify that the `timer.running` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |
| 71 | NegSem_2002_TheAltStatement_071 | verify that the `read` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 72 | NegSem_2002_TheAltStatement_072 | verify that the `activate` operation cannot be used in `alt` branch events (in templates) | Clause 20.2 | m | n |
| 73 | NegSem_2002_TheAltStatement_073 | verify that the `create` operation cannot be used in `alt` branch events (in inline template) | Clause 20.2 | m | n |
| 74 | NegSem_2002_TheAltStatement_074 | verify that the `component.running` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 75 | NegSem_2002_TheAltStatement_075 | verify that the `alive` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 76 | NegSem_2002_TheAltStatement_076 | verify that the `checkstate` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 77 | NegSem_2002_TheAltStatement_077 | verify that the `timer.running` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 78 | NegSem_2002_TheAltStatement_078 | verify that the `read` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 79 | NegSem_2002_TheAltStatement_079 | verify that the `activate` operation cannot be used in `alt` branch events (in template parameters) | Clause 20.2 | m | n |
| 80 | NegSem_2002_TheAltStatement_080 | verify that the `create` operation cannot be used in `alt` branch events (in inline template) | Clause 20.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 81 | NegSem_2002_TheAltStatement_081 | verify that the `component.running` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 82 | NegSem_2002_TheAltStatement_082 | verify that the `alive` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 83 | NegSem_2002_TheAltStatement_083 | verify that the `checkstate` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 84 | NegSem_2002_TheAltStatement_084 | verify that the `timer.running` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 85 | NegSem_2002_TheAltStatement_085 | verify that the `read` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 86 | NegSem_2002_TheAltStatement_086 | verify that the `activate` operation cannot be used in altstep declarations | Clause 20.2 | m | n |
| 87 | Sem_2002_TheAltStatement_001 | The `alt`-statement works as expected (loopback case). | Clause 20.2 | m | y |
| 88 | Sem_2002_TheAltStatement_002 | The `alt`-statement with a guard works as expected (loopback case). | Clause 20.2 | m | y |
| 89 | Sem_2002_TheAltStatement_003 | The `alt`-statement processes the alternatives in order (loopback case). | Clause 20.2 | m | y |
| 90 | Sem_2002_TheAltStatement_004 | Activated defaults are processed in the reverse order (loopback case). | Clause 20.2 | m | y |
| 91 | Sem_2002_TheAltStatement_005 | The else branch is executed when nothing else matched (loopback case). | Clause 20.2 | m | y |
| 92 | Sem_2002_TheAltStatement_006 | An altstep invocation works as expected (loopback case). | Clause 20.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 93 | Sem_2002_TheAltStatement_007 | An altstep invocation works as expected and that the `optional` statement block is executed after the altstep staatement block (loopback case). | Clause 20.2 | m | y |
| 94 | Sem_2002_TheAltStatement_008 | The done-block in an `alt` -statement is triggered as expected (loopback case). | Clause 20.2 | m | y |
| 95 | Sem_2002_TheAltStatement_009 | The killed-block in an `alt` -statement is triggered as expected when the component is killed (loopback case). | Clause 20.2 | m | y |
| 96 | Sem_2002_TheAltStatement_010 | The timeout branch is taken as expected (loopback case). | Clause 20.2 | m | y |
| 97 | Sem_2002_TheAltStatement_011 | The behaviour continues after the `alt`-statement (loopback case). | Clause 20.2 | m | y |
| 98 | Sem_2002_TheAltStatement_012 | `alt` statements are correctly handled for dynamically mapped ports | Clause 20.2 | m | y |
| 99 | Sem_2002_TheAltStatement_013 | `alt` statements are correctly handled for dynamically mapped ports | Clause 20.2 | m | y |
| 100 | Sem_2002_TheAltStatement_014 | no default activation after else | Clause 20.2 | m | y |

## 5.118. The `repeat` statement

*Table 117. The `repeat` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2003_the_repeat_statement_001 | The IUT correctly processes `repeat` statements | Clause 20.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_2003_the_repeat_statement_001 | The IUT correctly processes `repeat` statements | Clause 20.3 | m | y |
| 3 | Sem_2003_the_repeat_statement_002 | repeat in procedure call block | Clause 20.3 | m | n |
| 4 | Sem_2003_the_repeat_statement_003 | repeat in alstep branch of `alt` statements | Clause 20.3 | m | y |
| 5 | Sem_2003_the_repeat_statement_004 | repeat in executed default | Clause 20.3 | m | y |

## 5.119. The `interleave` statement

*Table 118. The `interleave` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2004_InterleaveStatement_001 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | n |
| 2 | NegSem_2004_InterleaveStatement_002 | while loop inside interleave | Clause 20.4 | m | n |
| 3 | NegSem_2004_InterleaveStatement_003 | do-while loop inside interleave | Clause 20.4 | m | n |
| 4 | NegSem_2004_InterleaveStatement_004 | goto inside interleave | Clause 20.4 | m | y |
| 5 | NegSem_2004_InterleaveStatement_005 | activate call inside interleave | Clause 20.4 | m | n |
| 6 | NegSem_2004_InterleaveStatement_006 | deactivate call inside interleave | Clause 20.4 | m | n |
| 7 | NegSem_2004_InterleaveStatement_007 | stop inside interleave | Clause 20.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_2004_InterleaveStatement_008 | repeat inside interleave | Clause 20.4 | m | y |
| 9 | NegSem_2004_InterleaveStatement_009 | return inside interleave | Clause 20.4 | m | y |
| 10 | NegSem_2004_InterleaveStatement_010 | explicit altstep call inside interleave | Clause 20.4 | m | y |
| 11 | NegSem_2004_InterleaveStatement_011 | direct function call containing `reception` statement inside interleave | Clause 20.4 | m | n |
| 12 | NegSem_2004_InterleaveStatement_012 | indirect function call containing `reception` statement inside interleave | Clause 20.4 | m | n |
| 13 | NegSyn_2004_InterleaveStatement_001 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | y |
| 14 | NegSyn_2004_InterleaveStatement_002 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | y |
| 15 | Sem_2004_InterleaveStatement_001 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | y |
| 16 | Sem_2004_InterleaveStatement_002 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | y |
| 17 | Sem_2004_InterleaveStatement_003 | while loop inside interleave | Clause 20.4 | m | y |
| 18 | Syn_2004_InterleaveStatement_001 | Validate that `interleave` statements are properly handled. | Clause 20.4 | m | y |

# 5.120. The default mechanism

*Table 119. The default mechanism*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_200501_the_default_mechanism_001 | verify unsuccessful default termination | Clause 20.5.1 | m | y |
| 2 | Sem_200501_the_default_mechanism_001 | verify that activated default is invoked | Clause 20.5.1 | m | y |
| 3 | Sem_200501_the_default_mechanism_002 | verify that default are processed in interleave | Clause 20.5.1 | m | y |
| 4 | Sem_200501_the_default_mechanism_003 | verify than default are processed in interleave | Clause 20.5.1 | m | y |
| 5 | Sem_200501_the_default_mechanism_004 | verify that default processing order is correct | Clause 20.5.1 | m | y |
| 6 | Sem_200501_the_default_mechanism_005 | verify that default processing order is correct | Clause 20.5.1 | m | y |
| 7 | Sem_200501_the_default_mechanism_006 | verify repeat command `behaviour` in invoked default | Clause 20.5.1 | m | y |
| 8 | Sem_200501_the_default_mechanism_007 | verify break command `behaviour` in invoked default | Clause 20.5.1 | m | y |
| 9 | Sem_200501_the_default_mechanism_008 | verify stop command `behaviour` in invoked default | Clause 20.5.1 | m | y |
| 10 | NegSem_200503_the_deactivate_operation_001 | verify that deactivate deactivated default causes error | Clause 20.5.3 | m | n |
| 11 | NegSem_200503_the_deactivate_operation_002 | verify that deactivate uninitialized default causes error | Clause 20.5.3 | m | y |
| 12 | NegSem_200503_the_deactivate_operation_003 | verify that error is generated when deactivated reference is on incorrect type | Clause 20.5.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | Sem_200503_the_deactivate_operation_001 | verify that deactivate removes default from list of defaults | Clause 20.5.3 | m | y |
| 14 | Sem_200503_the_deactivate_operation_002 | verify that deactivate removes default from list of defaults | Clause 20.5.3 | m | y |
| 15 | Sem_200503_the_deactivate_operation_003 | verify that deactivate without parameter clear list of defaults | Clause 20.5.3 | m | y |
| 16 | Sem_200503_the_deactivate_operation_004 | verify that deactivate `null` works correctly | Clause 20.5.3 | m | y |

# 5.121. The `activate` operation

*Table 120. The `activate` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_200502_the_activate_operation_001 | verify error is generated if activated alstep `runs on` incompatible component | Clause 20.5.2 | m | y |
| 2 | NegSem_200502_the_activate_operation_002 | verify error is generated when passing local timer | Clause 20.5.2 | m | y |
| 3 | NegSem_200502_the_activate_operation_003 | verify error is generated when activating altstep with `out` parameters | Clause 20.5.2 | m | n |
| 4 | NegSem_200502_the_activate_operation_004 | verify error is generated when activating altstep with `inout` parameters | Clause 20.5.2 | m | n |
| 5 | NegSem_200502_the_activate_operation_005 | verify error is generated when activating function | Clause 20.5.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_200502_the_activate_operation_006 | local timer as a parameter of activated altstep in module control | Clause 20.5.2 | m | y |
| 7 | NegSem_200502_the_activate_operation_007 | local timer (referenced through timer parameter) as a parameter of activated altstep in module control | Clause 20.5.2 | m | y |
| 8 | Sem_200502_the_activate_operation_001 | verify that `activate` operation can be used as standalone statement | Clause 20.5.2 | m | y |
| 9 | Sem_200502_the_activate_operation_002 | verify that parameters are passed at activation time | Clause 20.5.2 | m | y |
| 10 | Sem_200502_the_activate_operation_003 | verify that passing component timer to activated altstep | Clause 20.5.2 | m | y |
| 11 | Sem_200502_the_activate_operation_004 | verify passing port parameter to activated altstep | Clause 20.5.2 | m | y |
| 12 | Sem_200502_the_activate_operation_005 | control block timer as a parameter of activated altstep | Clause 20.5.2 | m | n |
| 13 | Sem_200502_the_activate_operation_006 | control block timer (referenced through timer parameter) as a parameter of activated altstep | Clause 20.5.2 | m | n |

## 5.122. `Connection` operations

*Table 121. `Connection` operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2101_TopLevel_001 | Verify that `connect` operation cannot contain a system port | Clause 21.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | NegSem_2101_TopLevel_002 | Verify that `map` operation fails if both operands are component ports | Clause 21.1 | m | y |

## 5.123. The `connect` and `map` operations

*Table 122. The `connect` and `map` operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210101_connect_and_map_operations_001 | Verify that `connect` operation rejects ports with incompatible message type lists | Clause 21.1.1 | m | y |
| 2 | NegSem_210101_connect_and_map_operations_002 | Verify that `connect` operation rejects ports with only partially compatible message type lists | Clause 21.1.1 | m | y |
| 3 | NegSem_210101_connect_and_map_operations_003 | Verify that `map` operation rejects ports with incompatible message type lists | Clause 21.1.1 | m | y |
| 4 | NegSem_210101_connect_and_map_operations_004 | Verify that `connect` operation rejects ports with only partially compatible message type lists | Clause 21.1.1 | m | y |
| 5 | NegSem_210101_connect_and_map_operations_005 | Verify that `map` parameters cannot be used when not declared in the port type | Clause 21.1.1 | m | n |
| 6 | NegSem_210101_connect_and_map_operations_006 | Verify that type incompatibility in `map` parameters is detected | Clause 21.1.1 | m | n |
| 7 | NegSem_210101_connect_and_map_operations_007 | Verify that parameter count mismatch in `map` param clause is detected | Clause 21.1.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_210101_connect_and_map_operations_008 | violation of strong typing rules for local ports in `connect` operations | Clause 21.1.1 | m | y |
| 9 | NegSem_210101_connect_and_map_operations_009 | violation of strong typing rules for MTC ports in `connect` operations | Clause 21.1.1 | m | n |
| 10 | NegSem_210101_connect_and_map_operations_010 | violation of strong typing rules for PTC ports in `connect` operations | Clause 21.1.1 | m | y |
| 11 | NegSem_210101_connect_and_map_operations_011 | violation of strong typing rules for local ports in `map` operations | Clause 21.1.1 | m | n |
| 12 | NegSem_210101_connect_and_map_operations_012 | violation of strong typing rules for MTC ports in `map` operations | Clause 21.1.1 | m | n |
| 13 | NegSem_210101_connect_and_map_operations_013 | violation of strong typing rules for PTC ports in `map` operations | Clause 21.1.1 | m | y |
| 14 | NegSem_210101_connect_and_map_operations_014 | violation of strong typing rules for system ports in `map` operations | Clause 21.1.1 | m | n |
| 15 | NegSem_210101_connect_operation_001 | The the IUT does not allows two output port connection | Clause 21.1.1 | m | y |
| 16 | NegSem_210101_connect_operation_002 | The the IUT does not allow connecting incompatible ports | Clause 21.1.1 | m | y |
| 17 | NegSem_210101_map_operation_001 | IUT cannot map input port with output port | Clause 21.1.1 | m | n |
| 18 | NegSem_210101_map_operation_002 | IUT cannot map input port with output port | Clause 21.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | Sem_210101_connect_and_map_operations_001 | `Connect` operation accepts ports with compatible message type list containing several types | Clause 21.1.1 | m | y |
| 20 | Sem_210101_connect_and_map_operations_002 | `Connect` operation accepts ports where `outlist` of the 1st port is a subset of `inlist` of the 2nd port | Clause 21.1.1 | m | y |
| 21 | Sem_210101_connect_and_map_operations_003 | `Connect` operation accepts ports where `outlist` of the 2nd port is a subset of `inlist` of the 1st port | Clause 21.1.1 | m | y |
| 22 | Sem_210101_connect_and_map_operations_004 | `Connect` operation accepts ports where `outlist` of both ports are subsets of `inlist` of the counterpart ports | Clause 21.1.1 | m | y |
| 23 | Sem_210101_connect_and_map_operations_005 | `Map` operation accepts ports with compatible message type list containing several types | Clause 21.1.1 | m | y |
| 24 | Sem_210101_connect_and_map_operations_006 | `Map` operation accepts ports with compatible message type list containing several types | Clause 21.1.1 | m | y |
| 25 | Sem_210101_connect_and_map_operations_007 | `Map` operation accepts ports with compatible message type list containing several types | Clause 21.1.1 | m | y |
| 26 | Sem_210101_connect_and_map_operations_008 | `Map` operation accepts ports with compatible message type list containing several types | Clause 21.1.1 | m | y |
| 27 | Sem_210101_connect_and_map_operations_009 | `Map` param statements are allowed in testcase block | Clause 21.1.1 | m | n |
| 28 | Sem_210101_connect_and_map_operations_010 | Verify that the param part can be skipped in `map` operations | Clause 21.1.1 | m | n |

# 5.124. The disconnect and unmap operations

*Table 123. The disconnect and unmap operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210102_disconnect_and_unmap_operations_001 | Verify that unmap operation cannot contain a system port reference | Clause 21.1.2 | m | y |
| 2 | NegSem_210102_disconnect_and_unmap_operations_002 | Verify that disconnecting all ports of all components is not possible in PTC | Clause 21.1.2 | m | n |
| 3 | NegSem_210102_disconnect_and_unmap_operations_003 | Verify that unmapping all ports of all components is not possible in PTC | Clause 21.1.2 | m | n |
| 4 | NegSem_210102_disconnect_and_unmap_operations_004 | Verify that unmap parameters cannot be used when not declared in the port type | Clause 21.1.2 | m | n |
| 5 | NegSem_210102_disconnect_and_unmap_operations_005 | Verify that type incompatibility in unmap parameters is detected | Clause 21.1.2 | m | n |
| 6 | NegSem_210102_disconnect_and_unmap_operations_006 | Verify that parameter count mismatch in unmap param clause is detected | Clause 21.1.2 | m | n |
| 7 | NegSem_210102_disconnect_and_unmap_operations_007 | Verify that the param clause cannot be used when unmap contains no system port reference | Clause 21.1.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_210102_disconnect_and_unmap_operations_008 | violation of strong typing rules for local ports in `disconnect` operations | Clause 21.1.2 | m | y |
| 9 | NegSem_210102_disconnect_and_unmap_operations_009 | violation of strong typing rules for MTC ports in `disconnect` operations | Clause 21.1.2 | m | n |
| 10 | NegSem_210102_disconnect_and_unmap_operations_010 | violation of strong typing rules for PTC ports in `disconnect` operations | Clause 21.1.2 | m | y |
| 11 | NegSem_210102_disconnect_and_unmap_operations_011 | violation of strong typing rules for local ports in `unmap` operations | Clause 21.1.2 | m | n |
| 12 | NegSem_210102_disconnect_and_unmap_operations_012 | violation of strong typing rules for MTC ports in `unmap` operations | Clause 21.1.2 | m | n |
| 13 | NegSem_210102_disconnect_and_unmap_operations_013 | violation of strong typing rules for PTC ports in `unmap` operations | Clause 21.1.2 | m | y |
| 14 | NegSem_210102_disconnect_and_unmap_operations_014 | violation of strong typing rules for system ports in `unmap` operations | Clause 21.1.2 | m | n |
| 15 | NegSem_210102_disconnect_operation_001 | Mapped port cannot disconnect | Clause 21.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 16 | Sem_210102_disconnect_and_unmap_operations_001 | `Disconnect` operation with two parameters works correctly | Clause 21.1.2 | m | y |
| 17 | Sem_210102_disconnect_and_unmap_operations_002 | `Disconnect` operation with one parameter works correctly | Clause 21.1.2 | m | n |
| 18 | Sem_210102_disconnect_and_unmap_operations_003 | `Disconnect` operation with all ports of a component works correctly | Clause 21.1.2 | m | n |
| 19 | Sem_210102_disconnect_and_unmap_operations_004 | `Disconnect` operation with no argument works correctly | Clause 21.1.2 | m | n |
| 20 | Sem_210102_disconnect_and_unmap_operations_005 | `Unmap` operation with one system port as a parameter works correctly | Clause 21.1.2 | m | n |
| 21 | Sem_210102_disconnect_and_unmap_operations_006 | `Unmap` operation with one component port as a parameter works correctly | Clause 21.1.2 | m | n |
| 22 | Sem_210102_disconnect_and_unmap_operations_007 | `Unmap` operation with all ports of a component works correctly | Clause 21.1.2 | m | n |
| 23 | Sem_210102_disconnect_and_unmap_operations_008 | `Unmap` operation with no parameters works correctly | Clause 21.1.2 | m | n |
| 24 | Sem_210102_disconnect_and_unmap_operations_009 | All component notation works correctly in `unmap` operations | Clause 21.1.2 | m | n |
| 25 | Sem_210102_disconnect_and_unmap_operations_010 | Verify that no error is generated when unmapping ports that are not mapped | Clause 21.1.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 26 | Sem_210102_disconnect_and_unmap_operations_011 | `Unmap` param statements are allowed in testcase block | Clause 21.1.2 | m | n |
| 27 | Sem_210102_disconnect_and_unmap_operations_012 | Verify that the param part can be skipped in `unmap` operations | Clause 21.1.2 | m | n |
| 28 | Sem_210102_disconnect_and_unmap_operations_013 | Verify that the param clause can be used when `unmap` contains a single system port parameter | Clause 21.1.2 | m | n |
| 29 | Sem_210102_disconnect_operation_001 | All component notation work correctly in `disconnect` operation | Clause 21.1.2 | m | n |
| 30 | Sem_210102_disconnect_operation_002 | Disconnect has no effect on components that are not connected | Clause 21.1.2 | m | y |
| 31 | Sem_210102_unmap_operation_001 | `Umnap` operation of a system and component port works correctly | Clause 21.1.2 | m | y |
| 32 | Sem_210102_unmap_operation_002 | `Umnap` operation of a component and system port works correctly | Clause 21.1.2 | m | y |

# 5.125. Test case operations

*Table 124. Test case operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2102_testcase_stop_001 | Stopping test case | Clause 21.2 | m | y |

# 5.126. The `create` operation

*Table 125. The `create` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210301_CreateOperation_001 | Named components on hosts are accepted | Clause 21.3.1 | m | y |
| 2 | NegSem_210301_CreateOperation_002 | Named components on hosts are accepted | Clause 21.3.1 | m | y |
| 3 | NegSem_210301_CreateOperation_003 | Named components on hosts are accepted | Clause 21.3.1 | m | y |
| 4 | Sem_210301_CreateOperation_001 | Unnamed components can be created | Clause 21.3.1 | m | y |
| 5 | Sem_210301_CreateOperation_002 | Named components can be created | Clause 21.3.1 | m | y |
| 6 | Sem_210301_CreateOperation_003 | Unnamed alive components on hosts can be created | Clause 21.3.1 | m | y |
| 7 | Sem_210301_CreateOperation_004 | Named alive components can be created | Clause 21.3.1 | m | y |
| 8 | Syn_210301_CreateOperation_001 | Named components on hosts are accepted | Clause 21.3.1 | m | y |

## 5.127. The `start test` component operation

*Table 126. The `start` test component operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210302_Start_test_component_001 | Non-alive ptc cannot start again | Clause 21.3.2 | m | y |
| 2 | NegSem_210302_Start_test_component_002 | Only component type is allowed for ptc declaration | Clause 21.3.2 | m | y |
| 3 | NegSem_210302_Start_test_component_003 | altstep in test component `start` operation | Clause 21.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_210302_Start_test_component_004 | starting behaviour on already running non-alive component | Clause 21.3.2 | m | y |
| 5 | NegSem_210302_Start_test_component_005 | starting behaviour on already running non-alive component | Clause 21.3.2 | m | y |
| 6 | NegSem_210302_Start_test_component_006 | function invocation in the `start` operation doesn't return a component | Clause 21.3.2 | m | y |
| 7 | NegSem_210302_Start_test_component_007 | starting function with incompatible `runs on` clause | Clause 21.3.2 | m | y |
| 8 | NegSem_210302_Start_test_component_008 | passing port to started component function | Clause 21.3.2 | m | y |
| 9 | NegSem_210302_Start_test_component_009 | passing default to started component function | Clause 21.3.2 | m | y |
| 10 | NegSem_210302_Start_test_component_010 | passing timer to started component function | Clause 21.3.2 | m | y |
| 11 | NegSem_210302_Start_test_component_011 | passing structured value containing ports to started component function | Clause 21.3.2 | m | y |
| 12 | NegSem_210302_Start_test_component_012 | passing default to started component function | Clause 21.3.2 | m | y |
| 13 | Sem_210302_Start_test_component_001 | Alive test components are allowed to start another function | Clause 21.3.2 | m | y |
| 14 | Sem_210302_Start_test_component_002 | component variable reference in `start` operation | Clause 21.3.2 | m | y |
| 15 | Sem_210302_Start_test_component_003 | test component as a result of function invocation in `start` operation | Clause 21.3.2 | m | y |
| 16 | Sem_210302_Start_test_component_004 | component variable value reuse in alive component | Clause 21.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 17 | Sem_210302_Start_test_component_005 | timer reuse in alive component | Clause 21.3.2 | m | y |
| 18 | Sem_210302_Start_test_component_006 | port reuse in alive component | Clause 21.3.2 | m | y |
| 19 | Sem_210302_Start_test_component_007 | verdict value reuse in alive component | Clause 21.3.2 | m | y |
| 20 | Sem_210302_Start_test_component_008 | timer reuse in alive component | Clause 21.3.2 | m | y |
| 21 | Sem_210302_Start_test_component_009 | deactivation of defaults in alive components | Clause 21.3.2 | m | n |
| 22 | Sem_210302_Start_test_component_010 | starting function with compatible `runs on` clause | Clause 21.3.2 | m | y |
| 23 | Sem_210302_Start_test_component_011 | altstep in test component `start` operation | Clause 21.3.2 | m | n |
| 24 | Sem_210302_Start_test_component_012 | `start` operation works with parametered altsteps | Clause 21.3.2 | m | n |
| 25 | Sem_210302_Start_test_component_013 | `inout` parameters will be passed to the function by value, i.e. like `in` -parameters | Clause 21.3.2 | m | y |
| 26 | Sem_210302_Start_test_component_014 | `inout` parameters will be passed to the function by value, i.e. like `in` -parameters | Clause 21.3.2 | m | y |

# 5.128. The `stop` test behaviour operation

*Table 127. The `stop` test behaviour operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210303_Stop_test_component_001 | restarting explicitly stopped non-alive component | Clause 21.3.3 | m | y |
| 2 | NegSem_210303_Stop_test_component_002 | stopping all PTCs from a PTC | Clause 21.3.3 | m | y |
| 3 | NegSem_210303_Stop_test_component_003 | applying `stop` operation to a variable of a different than component type | Clause 21.3.3 | m | y |
| 4 | NegSem_210303_Stop_test_component_004 | applying `stop` operation to a function call result of a different than component type | Clause 21.3.3 | m | y |
| 5 | Sem_210303_Stop_test_component_001 | `Component.stop` causes the stopping of the target component. | Clause 21.3.3 | m | y |
| 6 | Sem_210303_Stop_test_component_002 | `Self.stop` stops current component | Clause 21.3.3 | m | y |
| 7 | Sem_210303_Stop_test_component_003 | stopping MTC from PTC | Clause 21.3.3 | m | y |
| 8 | Sem_210303_Stop_test_component_004 | `stop.self` in MTC | Clause 21.3.3 | m | y |
| 9 | Sem_210303_Stop_test_component_005 | alive component restart after explicit stop | Clause 21.3.3 | m | y |
| 10 | Sem_210303_Stop_test_component_006 | component variable value reuse in alive component after explicit stop | Clause 21.3.3 | m | y |
| 11 | Sem_210303_Stop_test_component_007 | timer reuse in alive component after explicit stop | Clause 21.3.3 | m | y |
| 12 | Sem_210303_Stop_test_component_008 | port reuse in alive component after explicit stop | Clause 21.3.3 | m | y |
| 13 | Sem_210303_Stop_test_component_009 | verdict value reuse in alive component after explicit stop | Clause 21.3.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_210303_Stop_test_component_010 | deactivation of defaults in alive components after explicit stop | Clause 21.3.3 | m | n |
| 15 | Sem_210303_Stop_test_component_011 | stopping all PTCs | Clause 21.3.3 | m | y |

## 5.129. The `kill` test component operation

*Table 128. The `kill` test component operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210304_kill_test_component_001 | restarting explicitly killed non-alive component | Clause 21.3.4 | m | y |
| 15 | NegSem_210304_kill_test_component_002 | restarting explicitly killed alive component | Clause 21.3.4 | m | y |
| 2 | NegSem_210304_kill_test_component_003 | killing all PTCs from a PTC | Clause 21.3.4 | m | y |
| 3 | NegSem_210304_kill_test_component_004 | applying `kill` operation to a variable of a different than component type | Clause 21.3.4 | m | y |
| 4 | NegSem_210304_kill_test_component_005 | applying `kill` operation to a function call result of a different than component type | Clause 21.3.4 | m | y |
| 5 | Sem_210304_kill_test_component_001 | Kill operator stops a non alive test components. | Clause 21.3.4 | m | y |
| 6 | Sem_210304_kill_test_component_002 | All component kill stop all ptcs | Clause 21.3.4 | m | y |
| 7 | Sem_210304_kill_test_component_003 | Kill operator stops only non alive test components | Clause 21.3.4 | m | y |
| 8 | Sem_210304_kill_test_component_004 | Self kill called in a functions stops non alive test comp. | Clause 21.3.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | Sem_210304_kill_test_component_005 | standalone kill in alive PTC | Clause 21.3.4 | m | y |
| 10 | Sem_210304_kill_test_component_006 | killing MTC from PTC | Clause 21.3.4 | m | y |

## 5.130. The `alive` operation

*Table 129. The `alive` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210305_alive_operation_001 | Verify that error occurs when any from alive is applied to single component | Clause 21.3.5 | m | y |
| 2 | NegSem_210305_alive_operation_002 | Verify that error occurs when any from alive is applied to 1D array and index target is array | Clause 21.3.5 | m | y |
| 3 | NegSem_210305_alive_operation_003 | Verify that error occurs when any from alive is applied to 1D array and index target has wrong type | Clause 21.3.5 | m | y |
| 4 | NegSem_210305_alive_operation_004 | Verify that any from alive index redirection for multi-D arrays requires arrays of correct size | Clause 21.3.5 | m | y |
| 5 | NegSem_210305_alive_operation_005 | Verify that any from alive index redirection for multi-D arrays requires arrays | Clause 21.3.5 | m | y |
| 6 | NegSem_210305_alive_operation_006 | partially initialized array in any from `ComponentArrayRef.alive` | Clause 21.3.5 | m | n |
| 7 | NegSyn_210305_alive_operation_001 | Verify that error occurs when using index redirection in `component.alive` operation | Clause 21.3.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSyn_210305_alive_operation_002 | Verify that error occurs when using index redirection in any `component.alive` operation | Clause 21.3.5 | m | y |
| 9 | NegSyn_210305_alive_operation_003 | Verify that error occurs when using index redirection in all `component.alive` operation | Clause 21.3.5 | m | y |
| 10 | NegSyn_210305_alive_operation_004 | Verify that error occurs when using index redirection in function `instance.alive` operation | Clause 21.3.5 | m | y |
| 11 | Sem_210305_alive_operation_001 | Testing alive operator with an alive test component | Clause 21.3.5 | m | y |
| 12 | Sem_210305_alive_operation_002 | Test all component alive operator with alive test components | Clause 21.3.5 | m | y |
| 13 | Sem_210305_alive_operation_003 | Alive operator gives a correct boolean result | Clause 21.3.5 | m | y |
| 14 | Sem_210305_alive_operation_004 | Test any component alive operator with multiple test components | Clause 21.3.5 | m | y |
| 15 | Sem_210305_alive_operation_005 | Verify that any from alive returns false if no component is alive | Clause 21.3.5 | m | y |
| 16 | Sem_210305_alive_operation_006 | Verify that any from alive returns true if at least one component is inactive | Clause 21.3.5 | m | y |
| 17 | Sem_210305_alive_operation_007 | Verify that any from alive returns true if at least one component is running | Clause 21.3.5 | m | y |
| 18 | Sem_210305_alive_operation_008 | Verify that any from alive doesn't assign index when no component is alive | Clause 21.3.5 | m | y |
| 19 | Sem_210305_alive_operation_009 | Verify that any from alive assigns index | Clause 21.3.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 20 | Sem_210305_alive_operation_010 | Verify that any from alive can be used inside expressions | Clause 21.3.5 | m | y |
| 21 | Sem_210305_alive_operation_011 | Verify that any from alive index redirection works for multidimensional arrays | Clause 21.3.5 | m | y |
| 22 | Sem_210305_alive_operation_012 | Verify that any from alive doesn't change index variable when no component is alive | Clause 21.3.5 | m | y |
| 23 | Sem_210305_alive_operation_013 | Verify any from alive index redirection to lazy variable | Clause 21.3.5 | m | y |
| 24 | Sem_210305_alive_operation_014 | Verify any from alive index redirection to fuzzy variable | Clause 21.3.5 | m | y |
| 25 | Sem_210305_alive_operation_015 | Alive applied on the `mtc` the operation returns true | Clause 21.3.5 | m | n |

# 5.131. The `running` operation

*Table 130. The `running` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210306_running_operation_001 | Verify that error occurs when any from running is applied to single component | Clause 21.3.6 | m | y |
| 2 | NegSem_210306_running_operation_002 | Verify that error occurs when any from running is applied to 1D array and index target is array | Clause 21.3.6 | m | y |
| 3 | NegSem_210306_running_operation_003 | Verify that error occurs when any from running is applied to 1D array and index target has wrong type | Clause 21.3.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_210306_running_operation_004 | Verify that any from running index redirection for multi-D arrays requires arrays of correct size | Clause 21.3.6 | m | y |
| 5 | NegSem_210306_running_operation_005 | Verify that any from running index redirection for multi-D arrays requires arrays | Clause 21.3.6 | m | y |
| 6 | NegSem_210306_running_operation_006 | partially initialized array in any from `ComponentArrayRef.running` | Clause 21.3.6 | m | n |
| 7 | NegSyn_210306_running_operation_001 | Verify that error occurs when using index redirection in `component.running` operation | Clause 21.3.6 | m | y |
| 8 | NegSyn_210306_running_operation_002 | Verify that error occurs when using index redirection in any `component.running` operation | Clause 21.3.6 | m | y |
| 9 | NegSyn_210306_running_operation_003 | Verify that error occurs when using index redirection in all `component.running` operation | Clause 21.3.6 | m | y |
| 10 | NegSyn_210306_running_operation_004 | Verify that error occurs when using index redirection in function `instance.running` operation | Clause 21.3.6 | m | y |
| 11 | Sem_210306_running_operation_001 | Check that running operator provides information about test components. | Clause 21.3.6 | m | y |
| 12 | Sem_210306_running_operation_002 | Any component with running can check the status of the test components | Clause 21.3.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | Sem_210306_running_operation_003 | Verify that any from running returns false if no component is running | Clause 21.3.6 | m | y |
| 14 | Sem_210306_running_operation_004 | Verify that any from running returns true if at least one component is running | Clause 21.3.6 | m | y |
| 15 | Sem_210306_running_operation_005 | Verify that any from running doesn't assign index when no component is running | Clause 21.3.6 | m | y |
| 16 | Sem_210306_running_operation_006 | Verify that any from running doesn't change index variable when no component is running | Clause 21.3.6 | m | y |
| 17 | Sem_210306_running_operation_007 | Verify that any from running assigns index | Clause 21.3.6 | m | y |
| 18 | Sem_210306_running_operation_008 | Verify that any from running can be used inside expressions | Clause 21.3.6 | m | y |
| 19 | Sem_210306_running_operation_009 | Verify that any from running index redirection works for multidimensional arrays | Clause 21.3.6 | m | y |
| 20 | Sem_210306_running_operation_010 | Verify any from running index redirection to lazy variable | Clause 21.3.6 | m | y |
| 21 | Sem_210306_running_operation_011 | Verify any from running index redirection to fuzzy variable | Clause 21.3.6 | m | y |
| 22 | Sem_210306_running_operation_012 | Verify that all `component.running` produces true if some components haven't been started | Clause 21.3.6 | m | y |
| 23 | Sem_210306_running_operation_013 | Check that running operator provides information about `mtc` | Clause 21.3.6 | m | n |

# 5.132. The done operation

*Table 131. The done operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210307_done_operation_001 | Done operator can be used only for ptcs. | Clause 21.3.7 | m | y |
| 2 | NegSem_210307_done_operation_002 | Verify that error occurs when any from done is applied to single component | Clause 21.3.7 | m | y |
| 3 | NegSem_210307_done_operation_003 | Verify that error occurs when any from done is applied to 1D array and index target is array | Clause 21.3.7 | m | y |
| 4 | NegSem_210307_done_operation_004 | Verify that error occurs when any from done is applied to 1D array and index target has wrong type | Clause 21.3.7 | m | y |
| 5 | NegSem_210307_done_operation_005 | Verify that any from done index redirection for multi-D arrays requires arrays of correct size | Clause 21.3.7 | m | y |
| 6 | NegSem_210307_done_operation_006 | Verify that any from done index redirection for multi-D arrays requires arrays | Clause 21.3.7 | m | y |
| 7 | NegSem_210307_done_operation_007 | variable of incorrect type used for storing verdict in `done` operation | Clause 21.3.7 | m | n |
| 8 | NegSem_210307_done_operation_008 | storing verdict in any `component.done` operation | Clause 21.3.7 | m | n |
| 9 | NegSem_210307_done_operation_009 | storing verdict in all `component.done` operation | Clause 21.3.7 | m | n |
| 10 | NegSem_210307_done_operation_010 | partially initialized array in any from `ComponentArrayRef.done` | Clause 21.3.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | NegSyn_210307_done_operation_001 | Verify that error occurs when using index redirection in `component.done` operation | Clause 21.3.7 | m | y |
| 12 | NegSyn_210307_done_operation_002 | Verify that error occurs when using index redirection in any `component.done` operation | Clause 21.3.7 | m | y |
| 13 | NegSyn_210307_done_operation_003 | Verify that error occurs when using index redirection in all `component.done` operation | Clause 21.3.7 | m | y |
| 14 | NegSyn_210307_done_operation_004 | Verify that error occurs when using index redirection in function `instance.done` operation | Clause 21.3.7 | m | y |
| 15 | Sem_210307_done_operation_001 | All component with done can check that at least one test component is not done | Clause 21.3.7 | m | y |
| 16 | Sem_210307_done_operation_002 | Verify that any from done is not triggered if no component has been started | Clause 21.3.7 | m | y |
| 17 | Sem_210307_done_operation_003 | Verify that any from done matches if at least one component is stopped or killed | Clause 21.3.7 | m | y |
| 18 | Sem_210307_done_operation_004 | Verify that any from done doesn't assign index when no component has been stopped or killed | Clause 21.3.7 | m | y |
| 19 | Sem_210307_done_operation_005 | Verify that any from done doesn't change index variable when no component has been stopped or killed | Clause 21.3.7 | m | y |
| 20 | Sem_210307_done_operation_006 | Verify that any from done assigns index | Clause 21.3.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 21 | Sem_210307_done_operation_007 | Verify that any from done is not triggered if all components are executing function | Clause 21.3.7 | m | y |
| 22 | Sem_210307_done_operation_008 | Verify that any from done index redirection works for multidimensional arrays | Clause 21.3.7 | m | y |
| 23 | Sem_210307_done_operation_009 | Verify any from done index redirection to lazy variable | Clause 21.3.7 | m | y |
| 24 | Sem_210307_done_operation_010 | Verify any from done index redirection to fuzzy variable | Clause 21.3.7 | m | y |
| 25 | Sem_210307_done_operation_011 | Verify that all `component.done` produces true if some components haven't been started | Clause 21.3.7 | m | y |
| 26 | Sem_210307_done_operation_012 | storing verdict in `done` operation | Clause 21.3.7 | m | n |

## 5.133. The `killed` operation

*Table 132. The `killed` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_210308_killed_operation_001 | Killed operator is only valid for ptcs. | Clause 21.3.8 | m | y |
| 2 | NegSem_210308_killed_operation_002 | Verify that error occurs when any from killed is applied to single component | Clause 21.3.8 | m | y |
| 3 | NegSem_210308_killed_operation_003 | Verify that error occurs when any from killed is applied to 1D array and index target is array | Clause 21.3.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_210308_killed_operation_004 | Verify that error occurs when any from killed is applied to 1D array and index target has wrong type | Clause 21.3.8 | m | y |
| 5 | NegSem_210308_killed_operation_005 | Verify that any from killed index redirection for multi-D arrays requires arrays of correct size | Clause 21.3.8 | m | y |
| 6 | NegSem_210308_killed_operation_006 | Verify that any from killed index redirection for multi-D arrays requires arrays | Clause 21.3.8 | m | y |
| 7 | NegSem_210308_killed_operation_007 | variable of incorrect type used for storing verdict in `killed` operation | Clause 21.3.8 | m | n |
| 8 | NegSem_210308_killed_operation_008 | storing verdict in any `component.killed` operation | Clause 21.3.8 | m | n |
| 9 | NegSem_210308_killed_operation_009 | storing verdict in all `component.killed` operation | Clause 21.3.8 | m | n |
| 10 | NegSem_210308_killed_operation_010 | partially initialized array in any from `ComponentArrayRef.killed` | Clause 21.3.8 | m | y |
| 11 | NegSyn_210308_killed_operation_001 | Verify that error occurs when using index redirection in `component.killed` operation | Clause 21.3.8 | m | y |
| 12 | NegSyn_210308_killed_operation_002 | Verify that error occurs when using index redirection in any `component.killed` operation | Clause 21.3.8 | m | y |
| 13 | NegSyn_210308_killed_operation_003 | Verify that error occurs when using index redirection in all `component.killed` operation | Clause 21.3.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | NegSyn_210308_killed_operation_004 | Verify that error occurs when using index redirection in function `instance.killed` operation | Clause 21.3.8 | m | y |
| 15 | Sem_210308_killed_operation_001 | All component kill can be checked with killed operator | Clause 21.3.8 | m | y |
| 16 | Sem_210308_killed_operation_002 | check that any component and killed operator can check that at least one test component is running or not | Clause 21.3.8 | m | y |
| 17 | Sem_210308_killed_operation_003 | The alive keyword is properly evaluated | Clause 21.3.8 | m | y |
| 18 | Sem_210308_killed_operation_004 | Verify that any from killed is not triggered if no component has been started | Clause 21.3.8 | m | y |
| 19 | Sem_210308_killed_operation_005 | Verify that any from killed matches if at least one component is stopped or killed | Clause 21.3.8 | m | y |
| 20 | Sem_210308_killed_operation_006 | Verify that any from killed doesn’t assign index when no component has been killed | Clause 21.3.8 | m | y |
| 21 | Sem_210308_killed_operation_007 | Verify that any from killed doesn’t change index variable when no component has been killed | Clause 21.3.8 | m | y |
| 22 | Sem_210308_killed_operation_008 | Verify that any from killed assigns index | Clause 21.3.8 | m | y |
| 23 | Sem_210308_killed_operation_009 | Verify that any from killed is not triggered if all components are executing function | Clause 21.3.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 24 | Sem_210308_killed_operation_010 | Verify that any from killed index redirection works for multidimensional arrays | Clause 21.3.8 | m | y |
| 25 | Sem_210308_killed_operation_011 | Verify any from killed index redirection to lazy variable | Clause 21.3.8 | m | y |
| 26 | Sem_210308_killed_operation_012 | Verify any from killed index redirection to fuzzy variable | Clause 21.3.8 | m | y |
| 27 | Sem_210308_killed_operation_013 | Verify that any from killed is not triggered if when alive component has stopped execution | Clause 21.3.8 | m | y |
| 28 | Sem_210308_killed_operation_014 | storing verdict in `killed` operation | Clause 21.3.8 | m | n |

## 5.134. The `send` operation

*Table 133. The `send` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220201_SendOperation_001 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 2 | NegSem_220201_SendOperation_002 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 3 | NegSem_220201_SendOperation_003 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 4 | NegSem_220201_SendOperation_004 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 5 | NegSem_220201_SendOperation_005 | missing to clause in case of one-to-many connections | Clause 22.2.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_220201_SendOperation_006 | partially initialized template | Clause 22.2.1 | m | y |
| 7 | NegSem_220201_SendOperation_007 | no type prefix in inline template | Clause 22.2.1 | m | y |
| 8 | NegSem_220201_SendOperation_008 | incompatible address value in `send` operation | Clause 22.2.1 | m | n |
| 9 | NegSem_220201_SendOperation_009 | `null` address in the to clause of `send` operation | Clause 22.2.1 | m | n |
| 10 | NegSem_220201_SendOperation_010 | `null` component in the to clause of `send` operation | Clause 22.2.1 | m | y |
| 11 | NegSem_220201_SendOperation_011 | `send` operation on disconnected and unmapped ports | Clause 22.2.1 | m | y |
| 12 | Sem_220201_SendOperation_001 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 13 | Sem_220201_SendOperation_002 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 14 | Sem_220201_SendOperation_003 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 15 | Sem_220201_SendOperation_004 | The IUT correctly handles message sending operations | Clause 22.2.1 | m | y |
| 16 | Sem_220201_SendOperation_005 | unicast `send` operation | Clause 22.2.1 | m | y |
| 17 | Sem_220201_SendOperation_006 | multicast `send` operation | Clause 22.2.1 | m | n |
| 18 | Sem_220201_SendOperation_007 | broadcast `send` operation | Clause 22.2.1 | m | n |

# 5.135. The `receive` operation

*Table 134. The `receive` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220202_ReceiveOperation_001 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | y |
| 2 | NegSem_220202_ReceiveOperation_002 | no type prefix in ambiguous inline template | Clause 22.2.2 | m | y |
| 3 | NegSem_220202_ReceiveOperation_003 | type mismatch in redirect value assignment | Clause 22.2.2 | m | y |
| 4 | NegSem_220202_ReceiveOperation_004 | type mismatch in redirect assignment of message fields | Clause 22.2.2 | m | y |
| 5 | NegSem_220202_ReceiveOperation_005 | applying `@decoded` to a forbidden field | Clause 22.2.2 | m | y |
| 6 | NegSem_220202_ReceiveOperation_006 | decoding error in `@decoded` redirect assignment | Clause 22.2.2 | m | y |
| 7 | NegSem_220202_ReceiveOperation_007 | invalid format value in `@decoded` redirect assignment | Clause 22.2.2 | m | y |
| 8 | NegSem_220202_ReceiveOperation_008 | value of wrong type in `@decoded` redirect assignment | Clause 22.2.2 | m | y |
| 9 | NegSem_220202_ReceiveOperation_009 | `encoding` parameter of `@decoded` redirect assignment applied to incorrect type | Clause 22.2.2 | m | y |
| 10 | NegSem_220202_ReceiveOperation_010 | attempting to store component name in redirect assignment | Clause 22.2.2 | m | y |
| 11 | NegSem_220202_ReceiveOperation_011 | attempting to receive a type missing from the port list | Clause 22.2.2 | m | y |
| 12 | NegSem_220202_ReceiveOperation_012 | value redirect assignment in receive any message statement | Clause 22.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | NegSem_220202_ReceiveOperation_013 | trying to store address when receiving on connected port | Clause 22.2.2 | m | n |
| 14 | NegSem_220202_ReceiveOperation_014 | type mismatch in sender redirect assignment | Clause 22.2.2 | m | y |
| 15 | NegSem_220202_ReceiveOperation_015 | `null` component reference in `from` clause of `receive` operation | Clause 22.2.2 | m | y |
| 16 | NegSem_220202_ReceiveOperation_016 | `null` address reference in `from` clause of `receive` operation | Clause 22.2.2 | m | n |
| 17 | NegSem_220202_ReceiveOperation_017 | index redirection in standard `port.receive` | Clause 22.2.2 | m | y |
| 18 | NegSem_220202_ReceiveOperation_018 | index redirection in any `port.receive` | Clause 22.2.2 | m | n |
| 19 | NegSem_220202_ReceiveOperation_019 | insufficient value range of variable in index redirection | Clause 22.2.2 | m | y |
| 20 | NegSem_220202_ReceiveOperation_020 | insufficient array dimension of variable in index redirection | Clause 22.2.2 | m | y |
| 21 | NegSem_220202_ReceiveOperation_021 | insufficient element value range of variable in index redirection | Clause 22.2.2 | m | y |
| 22 | NegSem_220202_ReceiveOperation_022 | incompatible `from` and `sender` clause | Clause 22.2.2 | m | y |
| 23 | NegSem_220202_ReceiveOperation_023 | incompatible `decmatch` and `@decoded` value redirect | Clause 22.2.2 | m | n |
| 24 | Sem_220202_ReceiveOperation_001 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | y |
| 25 | Sem_220202_ReceiveOperation_002 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 26 | Sem_220202_ReceiveOperation_003 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | y |
| 27 | Sem_220202_ReceiveOperation_004 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | n |
| 28 | Sem_220202_ReceiveOperation_005 | The IUT correctly handles message receiving operations | Clause 22.2.2 | m | n |
| 29 | Sem_220202_ReceiveOperation_006 | receive with a `from` clause (single item) | Clause 22.2.2 | m | y |
| 30 | Sem_220202_ReceiveOperation_007 | receive with a `from` clause (multiple items) | Clause 22.2.2 | m | y |
| 31 | Sem_220202_ReceiveOperation_008 | receive with a `from` clause (any component) | Clause 22.2.2 | m | n |
| 32 | Sem_220202_ReceiveOperation_009 | redirect assignment of message fields | Clause 22.2.2 | m | y |
| 33 | Sem_220202_ReceiveOperation_010 | redirect assignment of message fields | Clause 22.2.2 | m | y |
| 34 | Sem_220202_ReceiveOperation_011 | `@decoded` redirect assignment of a bitstring field | Clause 22.2.2 | m | y |
| 35 | Sem_220202_ReceiveOperation_012 | `@decoded` redirect assignment of a hexstring field | Clause 22.2.2 | m | y |
| 36 | Sem_220202_ReceiveOperation_013 | `@decoded` redirect assignment of an octetstring field | Clause 22.2.2 | m | y |
| 37 | Sem_220202_ReceiveOperation_014 | `@decoded` redirect assignment of a charstring field | Clause 22.2.2 | m | y |
| 38 | Sem_220202_ReceiveOperation_015 | `@decoded` redirect assignment of a universal charstring field | Clause 22.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 39 | Sem_220202_ReceiveOperation_016 | `@decoded` redirect assignment with `encoding` parameter | Clause 22.2.2 | m | n |
| 40 | Sem_220202_ReceiveOperation_017 | redirect assignment storing a component | Clause 22.2.2 | m | y |
| 41 | Sem_220202_ReceiveOperation_018 | redirect assignment storing an address | Clause 22.2.2 | m | n |
| 42 | Sem_220202_ReceiveOperation_019 | any from `port.receive` statement | Clause 22.2.2 | m | y |
| 43 | Sem_220202_ReceiveOperation_020 | single dimensional index redirect in any from `port.receive` statement | Clause 22.2.2 | m | y |
| 44 | Sem_220202_ReceiveOperation_021 | multidimensional index redirect in any from `port.receive` statement | Clause 22.2.2 | m | y |
| 45 | Sem_220202_ReceiveOperation_022 | standalone receive as a shorthand for `alt` statement | Clause 22.2.2 | m | y |
| 46 | Sem_220202_ReceiveOperation_023 | single dimensional index redirect in any from `port.receive` statement | Clause 22.2.2 | m | n |
| 47 | Sem_220202_ReceiveOperation_024 | lazy variable in value redirect | Clause 22.2.2 | m | y |
| 48 | Sem_220202_ReceiveOperation_025 | lazy variable in sender redirect | Clause 22.2.2 | m | n |
| 49 | Sem_220202_ReceiveOperation_026 | lazy variable in index redirect | Clause 22.2.2 | m | y |
| 50 | Sem_220202_ReceiveOperation_027 | fuzzy variable in value redirect | Clause 22.2.2 | m | y |
| 51 | Sem_220202_ReceiveOperation_028 | fuzzy variable in sender redirect | Clause 22.2.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 52 | Sem_220202_ReceiveOperation_029 | fuzzy variable in `@index` redirect | Clause 22.2.2 | m | y |
| 53 | Sem_220202_ReceiveOperation_030 | verify that a variable of a different but compatible type can be used in a redirect assignment | Clause 22.2.2 | m | y |

## 5.136. The `trigger` operation

*Table 135. The `trigger` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220203_TriggerOperation_001 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | y |
| 2 | NegSem_220203_TriggerOperation_002 | no type prefix in ambiguous inline template | Clause 22.2.3 | m | y |
| 3 | NegSem_220203_TriggerOperation_003 | type mismatch in redirect value assignment | Clause 22.2.3 | m | y |
| 4 | NegSem_220203_TriggerOperation_004 | type mismatch in redirect assignment of message fields | Clause 22.2.3 | m | y |
| 5 | NegSem_220203_TriggerOperation_005 | applying `@decoded` to a forbidden field | Clause 22.2.3 | m | y |
| 6 | NegSem_220203_TriggerOperation_006 | decoding error in `@decoded` redirect assignment | Clause 22.2.3 | m | y |
| 7 | NegSem_220203_TriggerOperation_007 | invalid format value in `@decoded` redirect assignment | Clause 22.2.3 | m | y |
| 8 | NegSem_220203_TriggerOperation_008 | value of wrong type in `@decoded` redirect assignment | Clause 22.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | NegSem_220203_TriggerOperation_009 | `encoding` parameter of `@decoded` redirect assignment applied to incorrect type | Clause 22.2.3 | m | y |
| 10 | NegSem_220203_TriggerOperation_010 | attempting to store component name in redirect assignment | Clause 22.2.3 | m | y |
| 11 | NegSem_220203_TriggerOperation_011 | attempting to receive a type missing from the port list | Clause 22.2.3 | m | y |
| 12 | NegSem_220203_TriggerOperation_012 | value redirect assignment in receive any message statement | Clause 22.2.3 | m | y |
| 13 | NegSem_220203_TriggerOperation_013 | trying to store address with `trigger` operation on connected port | Clause 22.2.3 | m | n |
| 14 | NegSem_220203_TriggerOperation_014 | type mismatch in sender redirect assignment | Clause 22.2.3 | m | y |
| 15 | NegSem_220203_TriggerOperation_015 | `null` component reference in `from` clause of `trigger` operation | Clause 22.2.3 | m | y |
| 16 | NegSem_220203_TriggerOperation_016 | `null` address reference in `from` clause of receive operation | Clause 22.2.3 | m | n |
| 17 | NegSem_220203_TriggerOperation_017 | index redirection in standard `port.trigger` | Clause 22.2.3 | m | y |
| 18 | NegSem_220203_TriggerOperation_018 | index redirection in any `port.receive` | Clause 22.2.3 | m | n |
| 19 | NegSem_220203_TriggerOperation_019 | insufficient value range of variable in index redirection | Clause 22.2.3 | m | y |
| 20 | NegSem_220203_TriggerOperation_020 | insufficient array dimension of variable in index redirection | Clause 22.2.3 | m | y |
| 21 | NegSem_220203_TriggerOperation_021 | insufficient element value range of variable in index redirection | Clause 22.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 22 | NegSem_220203_TriggerOperation_022 | incompatible from and sender clause | Clause 22.2.3 | m | y |
| 23 | NegSem_220203_TriggerOperation_023 | incompatible `decmatch` and `@decoded` value redirect | Clause 22.2.3 | m | n |
| 24 | Sem_220203_TriggerOperation_001 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | y |
| 25 | Sem_220203_TriggerOperation_002 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | y |
| 26 | Sem_220203_TriggerOperation_003 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | y |
| 27 | Sem_220203_TriggerOperation_004 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | n |
| 28 | Sem_220203_TriggerOperation_005 | The IUT correctly handles message `trigger` operations | Clause 22.2.3 | m | n |
| 29 | Sem_220203_TriggerOperation_006 | trigger with a `from` clause (single item) | Clause 22.2.3 | m | y |
| 30 | Sem_220203_TriggerOperation_007 | trigger with a `from` clause (multiple items) | Clause 22.2.3 | m | y |
| 31 | Sem_220203_TriggerOperation_008 | trigger with a `from` clause (any component) | Clause 22.2.3 | m | n |
| 32 | Sem_220203_TriggerOperation_009 | redirect assignment of message fields | Clause 22.2.3 | m | y |
| 33 | Sem_220203_TriggerOperation_010 | redirect assignment of message fields | Clause 22.2.3 | m | y |
| 34 | Sem_220203_TriggerOperation_011 | `@decoded` redirect assignment of a bitstring field | Clause 22.2.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 35 | Sem_220203_TriggerOperation_012 | `@decoded` redirect assignment of a hexstring field | Clause 22.2.3 | m | y |
| 36 | Sem_220203_TriggerOperation_013 | `@decoded` redirect assignment of an octetstring field | Clause 22.2.3 | m | y |
| 37 | Sem_220203_TriggerOperation_014 | `@decoded` redirect assignment of a charstring field | Clause 22.2.3 | m | y |
| 38 | Sem_220203_TriggerOperation_015 | `@decoded` redirect assignment of a universal charstring field | Clause 22.2.3 | m | y |
| 39 | Sem_220203_TriggerOperation_016 | `@decoded` redirect assignment with `encoding` parameter | Clause 22.2.3 | m | n |
| 40 | Sem_220203_TriggerOperation_017 | redirect assignment storing a component | Clause 22.2.3 | m | y |
| 41 | Sem_220203_TriggerOperation_018 | redirect assignment storing an address | Clause 22.2.3 | m | n |
| 42 | Sem_220203_TriggerOperation_019 | any from `port.trigger` statement | Clause 22.2.3 | m | y |
| 43 | Sem_220203_TriggerOperation_020 | single dimensional index redirect in any from `port.trigger` statement | Clause 22.2.3 | m | y |
| 44 | Sem_220203_TriggerOperation_021 | multidimensional index redirect in any from `port.trigger` statement | Clause 22.2.3 | m | y |
| 45 | Sem_220203_TriggerOperation_022 | standalone trigger as a shorthand for `alt` statement | Clause 22.2.3 | m | y |
| 46 | Sem_220203_TriggerOperation_023 | lazy variable in value redirect | Clause 22.2.3 | m | y |
| 47 | Sem_220203_TriggerOperation_024 | lazy variable in sender redirect | Clause 22.2.3 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 48 | Sem_220203_TriggerOperation_025 | lazy variable in index redirect | Clause 22.2.3 | m | y |
| 49 | Sem_220203_TriggerOperation_026 | fuzzy variable in value redirect | Clause 22.2.3 | m | y |
| 50 | Sem_220203_TriggerOperation_027 | fuzzy variable in sender redirect | Clause 22.2.3 | m | n |
| 51 | Sem_220203_TriggerOperation_028 | fuzzy variable in `@index` redirect | Clause 22.2.3 | m | y |

# 5.137. The `call` operation

*Table 136. The `call` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220301_CallOperation_001 | The IUT correctly handles procedure `call` operations | Clause 22.3.1 | m | y |
| 2 | NegSem_220301_CallOperation_002 | The IUT correctly procedure calls | Clause 22.3.1 | m | y |
| 3 | NegSem_220301_CallOperation_003 | `null` component in the `to` clause of the `call` operation | Clause 22.3.1 | m | y |
| 4 | NegSem_220301_CallOperation_004 | `null` component in the multicast list of the `to` clause of the `call` operation | Clause 22.3.1 | m | y |
| 5 | NegSem_220301_CallOperation_005 | incompatible template in the `to` clause of the `call` operation | Clause 22.3.1 | m | y |
| 6 | NegSem_220301_CallOperation_006 | verify that non-blocking calls cannot have a response and exception handling part | Clause 22.3.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | NegSem_220301_CallOperation_007 | verify that signature that are not listed in the port `inout` and out list cannot be used in `call` operations | Clause 22.3.1 | m | y |
| 8 | NegSem_220301_CallOperation_008 | verify that in parameters of a signature used in a `call` operation cannot contain matching symbols | Clause 22.3.1 | m | y |
| 9 | NegSem_220301_CallOperation_009 | verify that in parameters of a signature used in a `call` operation cannot be omitted | Clause 22.3.1 | m | y |
| 10 | NegSem_220301_CallOperation_010 | verify that `inout` parameters of a signature used in a `call` operation cannot contain matching symbols | Clause 22.3.1 | m | y |
| 11 | NegSem_220301_CallOperation_011 | verify that `inout` parameters of a signature used in a `call` operation cannot be omitted | Clause 22.3.1 | m | y |
| 12 | NegSem_220301_CallOperation_012 | missing `to` clause in case of one-to-many connections | Clause 22.3.1 | m | y |
| 13 | NegSem_220301_CallOperation_013 | verify that type mismatch error is issued for incorrect `call` timer values | Clause 22.3.1 | m | y |
| 14 | NegSem_220301_CallOperation_014 | verify that `getreply` signature mismatch in the response and exception handling causes an error | Clause 22.3.1 | m | y |
| 15 | NegSem_220301_CallOperation_015 | verify that exception signature mismatch in the response and exception handling causes an error | Clause 22.3.1 | m | y |
| 16 | NegSem_220301_CallOperation_016 | verify that forbidden calls cannot appear in response and exception handling part guards | Clause 22.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 17 | NegSem_220301_CallOperation_017 | verify that forbidden functions cannot appear in response and exception handling part guards | Clause 22.3.1 | m | n |
| 18 | NegSem_220301_CallOperation_018 | verify that non-blocking procedure calls cannot contain timeout values | Clause 22.3.1 | m | y |
| 19 | NegSem_220301_CallOperation_019 | verify that non-blocking procedure calls cannot contain `nowait` parameter | Clause 22.3.1 | m | y |
| 20 | NegSem_220301_CallOperation_020 | verify that calls cannot be used on disconnected ports | Clause 22.3.1 | m | y |
| 21 | NegSyn_220301_CallOperation_001 | verify that the response and exception handling part cannot contain an `else` clause | Clause 22.3.1 | m | y |
| 22 | NegSyn_220301_CallOperation_002 | verify that the response and exception handling part cannot contain an altstep | Clause 22.3.1 | m | y |
| 23 | Sem_220301_CallOperation_001 | The IUT correctly handles procedure `call` operations | Clause 22.3.1 | m | y |
| 24 | Sem_220301_CallOperation_002 | The IUT correctly handles procedure `call` operations | Clause 22.3.1 | m | y |
| 25 | Sem_220301_CallOperation_003 | The IUT correctly handles non-blocking procedure call | Clause 22.3.1 | m | y |
| 26 | Sem_220301_CallOperation_004 | The IUT correctly handles non-blocking procedure call | Clause 22.3.1 | m | y |
| 27 | Sem_220301_CallOperation_005 | The IUT correctly handles multiple client calls to the same server | Clause 22.3.1 | m | n |
| 28 | Sem_220301_CallOperation_006 | The IUT correctly handles broadcast/multicast procedure call | Clause 22.3.1 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 29 | Sem_220301_CallOperation_007 | The IUT correctly handles broadcast/multicast procedure call | Clause 22.3.1 | m | n |
| 30 | Sem_220301_CallOperation_008 | The IUT correctly handles blocking procedure call | Clause 22.3.1 | m | y |
| 31 | Sem_220301_CallOperation_009 | Verify that defaults are not executed in response and exception handling part of a `call` operation | Clause 22.3.1 | m | y |
| 32 | Sem_220301_CallOperation_010 | Blocking call with no timeout | Clause 22.3.1 | m | y |
| 33 | Sem_220301_CallOperation_011 | Blocking broadcast call with response and exception handling part and subsequent `alt` | Clause 22.3.1 | m | n |
| 34 | Sem_220301_CallOperation_012 | Blocking broadcast call with response and exception handling part handling all replies | Clause 22.3.1 | m | n |
| 35 | Sem_220301_CallOperation_013 | Blocking multicast call with response and exception handling part and subsequent `alt` | Clause 22.3.1 | m | n |
| 36 | Sem_220301_CallOperation_014 | Blocking multicast call with response and exception handling part handling all replies | Clause 22.3.1 | m | n |
| 37 | Sem_220301_CallOperation_015 | Non-blocking broadcast call | Clause 22.3.1 | m | n |
| 38 | Sem_220301_CallOperation_016 | Non-blocking multicast call | Clause 22.3.1 | m | n |
| 39 | Sem_220301_CallOperation_017 | Non-blocking unicast call | Clause 22.3.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 40 | Sem_220301_CallOperation_018 | Verify that `out` parameters of a signature used in a `call` operation can be omitted | Clause 22.3.1 | m | y |
| 41 | Sem_220301_CallOperation_019 | Verify that `out` parameters of a signature used in a `call` operation can contain matching symbols | Clause 22.3.1 | m | y |
| 42 | Sem_220301_CallOperation_020 | Verify that replies that are not related to the actual call are ignored in unqualified `getreply` statements | Clause 22.3.1 | m | n |
| 43 | Sem_220301_CallOperation_021 | Verify that exceptions that are not related to the actual call are ignored in unqualified `catch` statements | Clause 22.3.1 | m | n |

# 5.138. The `getcall` operation

*Table 137. The `getcall` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220302_GetcallOperation_001 | `Getcall` operations are only used on procedure based ports | Clause 22.3.2 | m | y |
| 2 | NegSem_220302_GetcallOperation_002 | `Getcall` operation does not allow value assignment | Clause 22.3.2 | m | y |
| 3 | NegSem_220302_GetcallOperation_003 | `Getcall` for any call does not allow param assignment | Clause 22.3.2 | m | y |
| 4 | NegSem_220302_GetcallOperation_004 | Verify that error occurs when any from `getcall` is applied to single port | Clause 22.3.2 | m | y |
| 5 | NegSem_220302_GetcallOperation_005 | Verify that error occurs when any from `getcall` is applied to 1D array and index target is array | Clause 22.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | NegSem_220302_GetcallOperation_006 | Verify that error occurs when any from `getcall` is applied to 1D array and index target has wrong type | Clause 22.3.2 | m | y |
| 7 | NegSem_220302_GetcallOperation_007 | Verify that any from `getcall` index redirection for multi-D arrays requires arrays of correct size | Clause 22.3.2 | m | y |
| 8 | NegSem_220302_GetcallOperation_008 | Verify that any from `getcall` index redirection for multi-D arrays requires arrays | Clause 22.3.2 | m | y |
| 9 | NegSem_220302_GetcallOperation_009 | `null` component in the `from` clause of the `getcall` operation | Clause 22.3.2 | m | y |
| 10 | NegSem_220302_GetcallOperation_010 | `null` component in the multicast list of the `from` clause of the `getcall` operation | Clause 22.3.2 | m | n |
| 11 | NegSem_220302_GetcallOperation_011 | applying `@decoded` to a forbidden field | Clause 22.3.2 | m | y |
| 12 | NegSem_220302_GetcallOperation_012 | decoding error in `@decoded` redirect assignment | Clause 22.3.2 | m | y |
| 13 | NegSem_220302_GetcallOperation_013 | invalid format value in `@decoded` redirect assignment | Clause 22.3.2 | m | y |
| 14 | NegSem_220302_GetcallOperation_014 | value of wrong type in `@decoded` redirect assignment | Clause 22.3.2 | m | y |
| 15 | NegSem_220302_GetcallOperation_015 | `encoding` parameter of `@decoded` redirect assignment applied to incorrect type | Clause 22.3.2 | m | y |
| 16 | NegSem_220302_GetcallOperation_016 | incompatible from and `sender` clause in `getcall` operation | Clause 22.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 17 | NegSem_220302_GetcallOperation_017 | incompatible `decmatch` and `@decoded` value redirect | Clause 22.3.2 | m | n |
| 18 | NegSem_220302_GetcallOperation_018 | incompatible template in the `from` clause of the `getcall` operation | Clause 22.3.2 | m | n |
| 19 | NegSem_220302_GetcallOperation_019 | trying to store an incompatible component value in the `sender` clause of a `getcall` operation | Clause 22.3.2 | m | n |
| 20 | NegSyn_220302_GetcallOperation_001 | Verify that error occurs when using index redirection in `port.getcall` operation | Clause 22.3.2 | m | y |
| 21 | NegSyn_220302_GetcallOperation_002 | Verify that error occurs when using index redirection in any `port.getcall` operation | Clause 22.3.2 | m | n |
| 22 | Sem_220302_GetcallOperation_001 | `Getcall` operations remove only matching procedure from the queue | Clause 22.3.2 | m | y |
| 23 | Sem_220302_GetcallOperation_002 | `Getcall` operations remove the matching procedure from the queue | Clause 22.3.2 | m | y |
| 24 | Sem_220302_GetcallOperation_003 | The `getcall` operation can be correctly restricted to a certain client | Clause 22.3.2 | m | y |
| 25 | Sem_220302_GetcallOperation_004 | The `getcall` operation can be correctly restricted to a certain client | Clause 22.3.2 | m | y |
| 26 | Sem_220302_GetcallOperation_005 | `Getcall` operations work with any port attribute | Clause 22.3.2 | m | n |
| 27 | Sem_220302_GetcallOperation_006 | Verify that any from `getcall` is not triggered if there hasn't been any call | Clause 22.3.2 | m | y |
| 28 | Sem_220302_GetcallOperation_007 | Verify that any from `getcall` matches if at least one port contains enqueued call | Clause 22.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 29 | Sem_220302_GetcallOperation_008 | Verify that any from `getcall` doesn't assign index when there's no suitable match | Clause 22.3.2 | m | y |
| 30 | Sem_220302_GetcallOperation_009 | Verify that any from `getcall` doesn't change index variable when no there's no suitable match | Clause 22.3.2 | m | y |
| 31 | Sem_220302_GetcallOperation_010 | Verify that any from done assigns index | Clause 22.3.2 | m | y |
| 32 | Sem_220302_GetcallOperation_011 | Verify that any from `getcall` index redirection works for multidimensional arrays | Clause 22.3.2 | m | y |
| 33 | Sem_220302_GetcallOperation_012 | Verify any from `getcall` index redirection to lazy variable | Clause 22.3.2 | m | y |
| 34 | Sem_220302_GetcallOperation_013 | Verify any from `getcall` index redirection to fuzzy variable | Clause 22.3.2 | m | y |
| 35 | Sem_220302_GetcallOperation_014 | `@decoded` redirect assignment of a bitstring field | Clause 22.3.2 | m | y |
| 36 | Sem_220302_GetcallOperation_015 | `@decoded` redirect assignment of a hexstring field | Clause 22.3.2 | m | y |
| 37 | Sem_220302_GetcallOperation_016 | `@decoded` redirect assignment of an octetstring field | Clause 22.3.2 | m | y |
| 38 | Sem_220302_GetcallOperation_017 | `@decoded` redirect assignment of a charstring field | Clause 22.3.2 | m | y |
| 39 | Sem_220302_GetcallOperation_018 | `@decoded` redirect assignment of a universal charstring field | Clause 22.3.2 | m | y |
| 40 | Sem_220302_GetcallOperation_019 | `@decoded` redirect assignment with `encoding` parameter | Clause 22.3.2 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 41 | Sem_220302_GetcallOperation_020 | `getcall` with a `from` clause (single item) | Clause 22.3.2 | m | y |
| 42 | Sem_220302_GetcallOperation_021 | `getcall` with a `from` clause (multiple items) | Clause 22.3.2 | m | y |
| 43 | Sem_220302_GetcallOperation_022 | `getcall` with a `from` clause (any component) | Clause 22.3.2 | m | n |

# 5.139. The `reply` operation

*Table 138. The `reply` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220303_ReplyOperation_001 | `Reply` operations are only used on procedure based ports | Clause 22.3.3 | m | y |
| 2 | NegSem_220303_ReplyOperation_002 | `null` component in the `to` clause of the `reply` operation | Clause 22.3.3 | m | y |
| 3 | NegSem_220303_ReplyOperation_003 | `null` component in the multicast list of the `to` clause of the `reply` operation | Clause 22.3.3 | m | n |
| 4 | NegSem_220303_ReplyOperation_004 | verify that `reply` operation cannot be used on a message port | Clause 22.3.3 | m | y |
| 5 | NegSem_220303_ReplyOperation_005 | verify that signature not listed in the port definition cannot be used in the `reply` operation | Clause 22.3.3 | m | y |
| 6 | NegSem_220303_ReplyOperation_006 | verify that matching symbols cannot be used in `out` signature parameters in `reply` operations | Clause 22.3.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | NegSem_220303_ReplyOperation_007 | verify that matching symbols cannot be used in `inout` signature parameters in `reply` operations | Clause 22.3.3 | m | y |
| 8 | NegSem_220303_ReplyOperation_008 | verify that error is issued for a missing `to` clause in a `reply` operation in case of one-to-many connections | Clause 22.3.3 | m | y |
| 9 | NegSem_220303_ReplyOperation_009 | verify that values that are not addresses or components cannot be used in the `to` clause of the `reply` operation | Clause 22.3.3 | m | y |
| 10 | NegSem_220303_ReplyOperation_010 | verify that `reply` operation on a disconnected port causes an error | Clause 22.3.3 | m | y |
| 11 | Sem_220303_ReplyOperation_001 | The IUT correctly handles reply to multiple clients on the same server | Clause 22.3.3 | m | n |
| 12 | Sem_220303_ReplyOperation_002 | The IUT correctly handles reply to multiple clients on the same server | Clause 22.3.3 | m | n |
| 13 | Sem_220303_ReplyOperation_003 | verify that functionality of a simple `reply` operation (implicit unicast, no return value) | Clause 22.3.3 | m | y |
| 14 | Sem_220303_ReplyOperation_004 | verify that functionality of a simple `reply` operation (explicit unicast, return value) | Clause 22.3.3 | m | y |
| 15 | Sem_220303_ReplyOperation_005 | verify that in signature parameters of `reply` operations can contain matching symbols | Clause 22.3.3 | m | y |

## 5.140. The `getreply` operation

*Table 139. The `getreply` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220304_getreply_operation_001 | Verify that error occurs when any from `getreply` is applied to single port | Clause 22.3.4 | m | y |
| 2 | NegSem_220304_getreply_operation_002 | Verify that error occurs when any from `getreply` is applied to 1D array and index target is array | Clause 22.3.4 | m | y |
| 3 | NegSem_220304_getreply_operation_003 | Verify that error occurs when any from `getreply` is applied to 1D array and index target has wrong type | Clause 22.3.4 | m | y |
| 4 | NegSem_220304_getreply_operation_004 | Verify that any from `getreply` index redirection for multi-D arrays requires arrays of correct size | Clause 22.3.4 | m | y |
| 5 | NegSem_220304_getreply_operation_005 | Verify that any from `getreply` index redirection for multi-D arrays requires arrays | Clause 22.3.4 | m | y |
| 6 | NegSem_220304_getreply_operation_006 | `null` component in the `from` clause of the `getreply` operation | Clause 22.3.4 | m | y |
| 7 | NegSem_220304_getreply_operation_007 | `null` component in the multicast list of the from clause of the `getreply` operation | Clause 22.3.4 | m | y |
| 8 | NegSem_220304_getreply_operation_008 | applying `@decoded` to a forbidden parameter field | Clause 22.3.4 | m | y |
| 9 | NegSem_220304_getreply_operation_009 | decoding error in `@decoded` redirect parameter assignment | Clause 22.3.4 | m | y |
| 10 | NegSem_220304_getreply_operation_010 | invalid format value in `@decoded` redirect parameter assignment | Clause 22.3.4 | m | y |
| 11 | NegSem_220304_getreply_operation_011 | value of wrong type in `@decoded` redirect parameter assignment | Clause 22.3.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 12 | NegSem_220304_getreply_operation_012 | encoding parameter of `@decoded` redirect parameter assignment applied to incorrect type | Clause 22.3.4 | m | y |
| 13 | NegSem_220304_getreply_operation_013 | incompatible `from` and `sender` clause in `getreply` operation | Clause 22.3.4 | m | y |
| 14 | NegSem_220304_getreply_operation_014 | incompatible `decmatch` and `@decoded` parameter redirect | Clause 22.3.4 | m | n |
| 15 | NegSem_220304_getreply_operation_015 | applying `@decoded` to a forbidden parameter field | Clause 22.3.4 | m | y |
| 16 | NegSem_220304_getreply_operation_016 | decoding error in `@decoded` redirect value assignment | Clause 22.3.4 | m | y |
| 17 | NegSem_220304_getreply_operation_017 | invalid format value in `@decoded` redirect value assignment | Clause 22.3.4 | m | y |
| 18 | NegSem_220304_getreply_operation_018 | value of wrong type in `@decoded` redirect value assignment | Clause 22.3.4 | m | y |
| 19 | NegSem_220304_getreply_operation_019 | encoding parameter of `@decoded` redirect value assignment applied to incorrect type | Clause 22.3.4 | m | y |
| 20 | NegSem_220304_getreply_operation_020 | incompatible `decmatch` and `@decoded` value redirect | Clause 22.3.4 | m | n |
| 21 | NegSem_220304_getreply_operation_021 | incompatible template in the `from` clause of the `getreply` operation | Clause 22.3.4 | m | y |
| 22 | NegSem_220304_getreply_operation_022 | trying to store an incompatible component value in the `sender` clause of a `getreply` operation | Clause 22.3.4 | m | n |
| 23 | NegSyn_220304_getreply_operation_001 | Verify that error occurs when using index redirection in `port.getreply` operation | Clause 22.3.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 24 | NegSyn_220304_getreply_operation_002 | Verify that error occurs when using index redirection in any `port.getreply` operation | Clause 22.3.4 | m | n |
| 25 | Sem_220304_getreply_operation_001 | Verify that any from `getreply` is not triggered if there hasn't been any reply | Clause 22.3.4 | m | y |
| 26 | Sem_220304_getreply_operation_002 | Verify that any from `getreply` matches if at least one port contains enqueued reply | Clause 22.3.4 | m | y |
| 27 | Sem_220304_getreply_operation_003 | Verify that any from `getreply` doesn't assign index when there's no suitable match | Clause 22.3.4 | m | y |
| 28 | Sem_220304_getreply_operation_004 | Verify that any from `getreply` doesn't change index variable when no there's no suitable match | Clause 22.3.4 | m | y |
| 29 | Sem_220304_getreply_operation_005 | Verify that any from done assigns index | Clause 22.3.4 | m | y |
| 30 | Sem_220304_getreply_operation_006 | Verify that any from `getreply` index redirection works for multidimensional arrays | Clause 22.3.4 | m | y |
| 31 | Sem_220304_getreply_operation_007 | Verify any from `getreply` index redirection to lazy variable | Clause 22.3.4 | m | y |
| 32 | Sem_220304_getreply_operation_008 | Verify any from `getreply` index redirection to fuzzy variable | Clause 22.3.4 | m | y |
| 33 | Sem_220304_getreply_operation_009 | `@decoded` redirect parameter assignment of a bitstring field | Clause 22.3.4 | m | y |
| 34 | Sem_220304_getreply_operation_010 | `@decoded` redirect parameter assignment of a hexstring field | Clause 22.3.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 35 | Sem_220304_getreply_operation_011 | `@decoded` redirect parameter assignment of an octetstring field | Clause 22.3.4 | m | y |
| 36 | Sem_220304_getreply_operation_012 | `@decoded` redirect parameter assignment of a charstring field | Clause 22.3.4 | m | y |
| 37 | Sem_220304_getreply_operation_013 | `@decoded` redirect parameter assignment of a universal charstring field | Clause 22.3.4 | m | y |
| 38 | Sem_220304_getreply_operation_014 | `@decoded` redirect parameter assignment with encoding parameter | Clause 22.3.4 | m | n |
| 39 | Sem_220304_getreply_operation_015 | `@decoded` redirect value assignment of a bitstring field | Clause 22.3.4 | m | y |
| 40 | Sem_220304_getreply_operation_016 | `@decoded` redirect value assignment of a hexstring field | Clause 22.3.4 | m | y |
| 41 | Sem_220304_getreply_operation_017 | `@decoded` redirect value assignment of an octetstring field | Clause 22.3.4 | m | y |
| 42 | Sem_220304_getreply_operation_018 | `@decoded` redirect value assignment of a charstring field | Clause 22.3.4 | m | y |
| 43 | Sem_220304_getreply_operation_019 | `@decoded` redirect value assignment of a universal charstring field | Clause 22.3.4 | m | y |
| 44 | Sem_220304_getreply_operation_020 | `@decoded` redirect value assignment with encoding parameter | Clause 22.3.4 | m | n |
| 45 | Sem_220304_getreply_operation_021 | `getreply` with a `from` clause (single item) | Clause 22.3.4 | m | y |
| 46 | Sem_220304_getreply_operation_022 | `getreply` with a `from` clause (multiple items) | Clause 22.3.4 | m | y |
| 47 | Sem_220304_getreply_operation_023 | `getreply` with a `from` clause (any component) | Clause 22.3.4 | m | n |

# 5.141. The `raise` operation

*Table 140. The `raise` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220305_raise_operation_001 | raised exception type not in the list of available exceptions | Clause 22.3.5 | m | y |
| 2 | NegSem_220305_raise_operation_002 | exception raised for a signature with no exception list | Clause 22.3.5 | m | y |
| 3 | NegSem_220305_raise_operation_003 | raised exception type is ambiguous | Clause 22.3.5 | m | y |
| 4 | NegSem_220305_raise_operation_004 | missing `to` clause in case of 1 to n connection | Clause 22.3.5 | m | y |
| 5 | NegSem_220305_raise_operation_005 | exception on a message port | Clause 22.3.5 | m | y |
| 6 | NegSem_220305_raise_operation_006 | exception procedure signature not in the port list | Clause 22.3.5 | m | y |
| 7 | NegSem_220305_raise_operation_007 | value of incorrect type in the `to` clause of the `raise` operation | Clause 22.3.5 | m | y |
| 8 | NegSem_220305_raise_operation_008 | `null` in the `to` clause of the `raise` operation | Clause 22.3.5 | m | y |
| 9 | NegSem_220305_raise_operation_009 | `raise` operation on disconnected and unmapped ports | Clause 22.3.5 | m | y |
| 10 | NegSem_220305_raise_operation_010 | exception template not conforming to template(value) restriction | Clause 22.3.5 | m | y |
| 11 | Sem_220305_raise_operation_001 | simple `raise` operation | Clause 22.3.5 | m | y |
| 12 | Sem_220305_raise_operation_002 | unicast `raise` operation | Clause 22.3.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 13 | Sem_220305_raise_operation_003 | broadcast `raise` operation | Clause 22.3.5 | m | n |
| 14 | Sem_220305_raise_operation_004 | multicast `raise` operation | Clause 22.3.5 | m | n |

## 5.142. The `catch` operation

*Table 141. The `catch` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_220306_catch_operation_001 | Verify that error occurs when any from catch is applied to single port | Clause 22.3.6 | m | y |
| 2 | NegSem_220306_catch_operation_002 | Verify that error occurs when any from catch is applied to 1D array and index target is array | Clause 22.3.6 | m | y |
| 3 | NegSem_220306_catch_operation_003 | Verify that error occurs when any from catch is applied to 1D array and index target has wrong type | Clause 22.3.6 | m | y |
| 4 | NegSem_220306_catch_operation_004 | Verify that any from catch index redirection for multi-D arrays requires arrays of correct size | Clause 22.3.6 | m | y |
| 5 | NegSem_220306_catch_operation_005 | Verify that any from catch index redirection for multi-D arrays requires arrays | Clause 22.3.6 | m | y |
| 6 | NegSem_220306_catch_operation_006 | `null` component in the `from` clause of the `catch` operation | Clause 22.3.6 | m | y |
| 7 | NegSem_220306_catch_operation_007 | `null` component in the multicast list of the `from` clause of the `catch` operation | Clause 22.3.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | NegSem_220306_catch_operation_008 | applying `@decoded` to a forbidden exception field | Clause 22.3.6 | m | y |
| 9 | NegSem_220306_catch_operation_009 | decoding error in `@decoded` redirect value assignment | Clause 22.3.6 | m | y |
| 10 | NegSem_220306_catch_operation_010 | invalid format value in `@decoded` redirect value assignment | Clause 22.3.6 | m | y |
| 11 | NegSem_220306_catch_operation_011 | value of wrong type in `@decoded` redirect value assignment | Clause 22.3.6 | m | y |
| 12 | NegSem_220306_catch_operation_012 | encoding parameter of `@decoded` redirect value assignment applied to incorrect type | Clause 22.3.6 | m | y |
| 13 | NegSem_220306_catch_operation_013 | incompatible `from` and `sender` clause in `catch` operation | Clause 22.3.6 | m | y |
| 14 | NegSem_220306_catch_operation_014 | incompatible `decmatch` and `@decoded` value redirect | Clause 22.3.6 | m | n |
| 15 | NegSem_220306_catch_operation_015 | incompatible template in the `from` clause of the `catch` operation | Clause 22.3.6 | m | y |
| 16 | NegSem_220306_catch_operation_016 | trying to store an incompatible component value in the `sender` clause of a `catch` operation | Clause 22.3.6 | m | n |
| 17 | NegSyn_220306_catch_operation_001 | Verify that error occurs when using index redirection in `port.catch` operation | Clause 22.3.6 | m | y |
| 18 | NegSyn_220306_catch_operation_002 | Verify that error occurs when using index redirection in any `port.catch` operation | Clause 22.3.6 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | NegSyn_220306_catch_operation_003 | Verify that error occurs when any from catch is applied to 1D array and index target has wrong type | Clause 22.3.6 | m | n |
| 20 | Sem_220306_catch_operation_001 | Verify that any from catch is not triggered if there hasn't been any exception | Clause 22.3.6 | m | y |
| 21 | Sem_220306_catch_operation_002 | Verify that any from catch matches if at least one port contains enqueued reply | Clause 22.3.6 | m | y |
| 22 | Sem_220306_catch_operation_003 | Verify that any from catch doesn't assign index when there's no suitable match | Clause 22.3.6 | m | y |
| 23 | Sem_220306_catch_operation_004 | Verify that any from catch doesn't change index variable when no there's no suitable match | Clause 22.3.6 | m | y |
| 24 | Sem_220306_catch_operation_005 | Verify that any from done assigns index | Clause 22.3.6 | m | y |
| 25 | Sem_220306_catch_operation_006 | Verify that any from catch index redirection works for multidimensional arrays | Clause 22.3.6 | m | y |
| 26 | Sem_220306_catch_operation_007 | Verify any from catch index redirection to lazy variable | Clause 22.3.6 | m | y |
| 27 | Sem_220306_catch_operation_008 | Verify any from catch index redirection to fuzzy variable | Clause 22.3.6 | m | y |
| 28 | Sem_220306_catch_operation_009 | `@decoded` redirect value assignment of a bitstring field | Clause 22.3.6 | m | y |
| 29 | Sem_220306_catch_operation_010 | `@decoded` redirect value assignment of a hexstring field | Clause 22.3.6 | m | y |
| 30 | Sem_220306_catch_operation_011 | `@decoded` redirect value assignment of an octetstring field | Clause 22.3.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 31 | Sem_220306_catch_operation_012 | `@decoded` redirect value assignment of a charstring field | Clause 22.3.6 | m | y |
| 32 | Sem_220306_catch_operation_013 | `@decoded` redirect value assignment of a universal charstring field | Clause 22.3.6 | m | y |
| 33 | Sem_220306_catch_operation_014 | `@decoded` redirect value assignment with encoding parameter | Clause 22.3.6 | m | n |
| 34 | Sem_220306_catch_operation_015 | catch with a `from` clause (single item) | Clause 22.3.6 | m | y |
| 35 | Sem_220306_catch_operation_016 | catch with a `from` clause (multiple items) | Clause 22.3.6 | m | y |
| 36 | Sem_220306_catch_operation_017 | catch with a `from` clause (any component) | Clause 22.3.6 | m | n |

## 5.143. The `check` operation

*Table 142. The `check` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2204_the_check_operation_001 | `null` component reference in `from` clause of `check` operation | Clause 22.4 | m | y |
| 2 | NegSem_2204_the_check_operation_002 | `null` address reference in `from` clause of `check` operation | Clause 22.4 | m | n |
| 3 | NegSem_2204_the_check_operation_003 | incompatible `from` and `sender` clause | Clause 22.4 | m | n |
| 4 | NegSem_2204_the_check_operation_004 | incompatible value in the `from` clause | Clause 22.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSem_2204_the_check_operation_005 | verify that a runtime error is generated if the real sender is incompatible with the variable in sender redirect assignment | Clause 22.4 | m | n |
| 6 | Sem_2204_the_check_operation_001 | Verify that `port.check(receive)` works correctly inside `alt` | Clause 22.4 | m | y |
| 7 | Sem_2204_the_check_operation_002 | Verify that `port.check(receive)` with assignment works correctly inside `alt` | Clause 22.4 | m | n |
| 8 | Sem_2204_the_check_operation_003 | Verify that `port.check(receive)` works correctly as standalone statement | Clause 22.4 | m | y |
| 9 | Sem_2204_the_check_operation_004 | Verify that `port.check(receive)` with assignment works correctly as standalone statement | Clause 22.4 | m | n |
| 10 | Sem_2204_the_check_operation_005 | Verify that any `port.check(receive)` works correctly inside `alt` | Clause 22.4 | m | y |
| 11 | Sem_2204_the_check_operation_006 | Verify that any `port.check(receive)` with assignment works correctly inside `alt` | Clause 22.4 | m | n |
| 12 | Sem_2204_the_check_operation_007 | Verify that any `port.check(receive)` works correctly as standalone statement | Clause 22.4 | m | y |
| 13 | Sem_2204_the_check_operation_008 | Verify that any `port.check(receive)` with assignment works correctly as standalone statement | Clause 22.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Sem_2204_the_check_operation_009 | Verify behaviour of `port.check(receive)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 15 | Sem_2204_the_check_operation_010 | Verify behaviour of `port.check(receive)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 16 | Sem_2204_the_check_operation_011 | Verify `port.check(receive)` behaviour in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 17 | Sem_2204_the_check_operation_012 | Verify behaviour of `port.check(receive)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 18 | Sem_2204_the_check_operation_013 | Verify any `port.check(receive)` behaviour in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 19 | Sem_2204_the_check_operation_014 | Verify behaviour of any `port.check(receive)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 20 | Sem_2204_the_check_operation_015 | Verify any `port.check(receive)` behaviour in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 21 | Sem_2204_the_check_operation_016 | Verify behaviour of any `port.check(receive)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 22 | Sem_2204_the_check_operation_017 | Verify behaviour of `port.check(receive)` in case of successful match inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 23 | Sem_2204_the_check_operation_018 | Verify behaviour of `port.check(receive)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | n |
| 24 | Sem_2204_the_check_operation_019 | Verify behaviour of `port.check(receive)` in case of successful match in standalone statement | Clause 22.4 | m | y |
| 25 | Sem_2204_the_check_operation_020 | Verify behaviour of `port.check(receive)` with assignment in case of successful match works correctly as standalone statement | Clause 22.4 | m | y |
| 26 | Sem_2204_the_check_operation_021 | Verify behaviour of any `port.check(receive)` in case of successful match inside `alt` | Clause 22.4 | m | n |
| 27 | Sem_2204_the_check_operation_022 | Verify behaviour of any `port.check(receive)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | n |
| 28 | Sem_2204_the_check_operation_023 | Verify behaviour of any `port.check(receive)` in case of successful match in standalone statement | Clause 22.4 | m | n |
| 29 | Sem_2204_the_check_operation_024 | Verify behaviour of any `port.check(receive)` with assignment in case of successful match works correctly as standalone statement | Clause 22.4 | m | n |
| 30 | Sem_2204_the_check_operation_025 | Verify that `port.check(getcall)` works correctly inside `alt` | Clause 22.4 | m | y |
| 31 | Sem_2204_the_check_operation_026 | Verify that `port.check(getcall)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 32 | Sem_2204_the_check_operation_027 | Verify that `port.check(getcall)` works correctly as standalone statement | Clause 22.4 | m | y |
| 33 | Sem_2204_the_check_operation_028 | Verify that `port.check(getcall)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 34 | Sem_2204_the_check_operation_029 | Verify that any `port.check(getcall)` works correctly inside `alt` | Clause 22.4 | m | y |
| 35 | Sem_2204_the_check_operation_030 | Verify that any `port.check(getcall)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 36 | Sem_2204_the_check_operation_031 | Verify that any `port.check(getcall)` works correctly as standalone statement | Clause 22.4 | m | y |
| 37 | Sem_2204_the_check_operation_032 | Verify that any `port.check(getcall)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 38 | Sem_2204_the_check_operation_033 | Verify behaviour of `port.check(getcall)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 39 | Sem_2204_the_check_operation_034 | Verify behaviour of `port.check(getcall)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 40 | Sem_2204_the_check_operation_035 | Verify behaviour of `port.check(getcall)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 41 | Sem_2204_the_check_operation_036 | Verify behaviour of `port.check(getcall)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 42 | Sem_2204_the_check_operation_037 | Verify behaviour of any `port.check(getcall)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 43 | Sem_2204_the_check_operation_038 | Verify behaviour of any `port.check(getcall)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 44 | Sem_2204_the_check_operation_039 | Verify behaviour of any `port.check(getcall)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 45 | Sem_2204_the_check_operation_040 | Verify behaviour of any `port.check(getcall)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 46 | Sem_2204_the_check_operation_041 | Verify behaviour of `port.check(getcall)` in case of successful match inside `alt` | Clause 22.4 | m | y |
| 47 | Sem_2204_the_check_operation_042 | Verify behaviour of `port.check(getcall)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | y |
| 48 | Sem_2204_the_check_operation_043 | Verify behaviour of `port.check(getcall)` in case of successful match in standalone statement | Clause 22.4 | m | y |
| 49 | Sem_2204_the_check_operation_044 | Verify behaviour of `port.check(getcall)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 50 | Sem_2204_the_check_operation_045 | Verify behaviour of any `port.check(getcall)` in case of successful match inside `alt` | Clause 22.4 | m | n |
| 51 | Sem_2204_the_check_operation_046 | Verify behaviour of any `port.check(getcall)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | n |
| 52 | Sem_2204_the_check_operation_047 | Verify behaviour of any `port.check(getcall)` in case of successful match in standalone statement | Clause 22.4 | m | n |
| 53 | Sem_2204_the_check_operation_048 | Verify behaviour of any `port.check(getcall)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | n |
| 54 | Sem_2204_the_check_operation_049 | Verify that `port.check(getreply)` works correctly inside `alt` | Clause 22.4 | m | y |
| 55 | Sem_2204_the_check_operation_050 | Verify that `port.check(getreply)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 56 | Sem_2204_the_check_operation_051 | Verify that `port.check(getreply)` works correctly as standalone statement | Clause 22.4 | m | y |
| 57 | Sem_2204_the_check_operation_052 | Verify that `port.check(getreply)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 58 | Sem_2204_the_check_operation_053 | Verify that any `port.check(getreply)` works correctly inside `alt` | Clause 22.4 | m | y |
| 59 | Sem_2204_the_check_operation_054 | Verify that any `port.check(getreply)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 60 | Sem_2204_the_check_operation_055 | Verify that any `port.check(getreply)` works correctly as standalone statement | Clause 22.4 | m | y |
| 61 | Sem_2204_the_check_operation_056 | Verify that any `port.check(getreply)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 62 | Sem_2204_the_check_operation_057 | Verify behaviour of `port.check(getreply)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 63 | Sem_2204_the_check_operation_058 | Verify behaviour of `port.check(getreply)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 64 | Sem_2204_the_check_operation_059 | Verify behaviour of `port.check(getreply)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 65 | Sem_2204_the_check_operation_060 | Verify behaviour of `port.check(getreply)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 66 | Sem_2204_the_check_operation_061 | Verify behaviour of any `port.check(getreply)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 67 | Sem_2204_the_check_operation_062 | Verify behaviour of any `port.check(getreply)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 68 | Sem_2204_the_check_operation_063 | Verify behaviour of any `port.check(getreply)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 69 | Sem_2204_the_check_operation_064 | Verify behaviour of any `port.check(getreply)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 70 | Sem_2204_the_check_operation_065 | Verify behaviour of `port.check(getreply)` in case of successful match inside `alt` | Clause 22.4 | m | y |
| 71 | Sem_2204_the_check_operation_066 | Verify behaviour of `port.check(getreply)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | y |
| 72 | Sem_2204_the_check_operation_067 | Verify behaviour of `port.check(getreply)` in case of successful match in standalone statement | Clause 22.4 | m | y |
| 73 | Sem_2204_the_check_operation_068 | Verify behaviour of `port.check(getreply)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | y |
| 74 | Sem_2204_the_check_operation_069 | Verify behaviour of any `port.check(getreply)` in case of successful match inside `alt` | Clause 22.4 | m | n |
| 75 | Sem_2204_the_check_operation_070 | Verify behaviour of any `port.check(getreply)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | n |
| 76 | Sem_2204_the_check_operation_071 | Verify behaviour of any `port.check(getreply)` in case of successful match in standalone statement | Clause 22.4 | m | n |
| 77 | Sem_2204_the_check_operation_072 | Verify behaviour of any `port.check(getreply)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 78 | Sem_2204_the_check_operation_073 | Verify that `port.check(catch)` works correctly inside `alt` | Clause 22.4 | m | y |
| 79 | Sem_2204_the_check_operation_074 | Verify that `port.check(catch)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 80 | Sem_2204_the_check_operation_075 | Verify that `port.check(catch)` works correctly as standalone statement | Clause 22.4 | m | y |
| 81 | Sem_2204_the_check_operation_076 | Verify that `port.check(catch)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 82 | Sem_2204_the_check_operation_077 | Verify that any `port.check(catch)` works correctly inside `alt` | Clause 22.4 | m | y |
| 83 | Sem_2204_the_check_operation_078 | Verify that any `port.check(catch)` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 84 | Sem_2204_the_check_operation_079 | Verify that any `port.check(catch)` works correctly as standalone statement | Clause 22.4 | m | y |
| 85 | Sem_2204_the_check_operation_080 | Verify that any `port.check(catch)` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 86 | Sem_2204_the_check_operation_081 | Verify behaviour of `port.check(catch)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 87 | Sem_2204_the_check_operation_082 | Verify behaviour of `port.check(catch)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 88 | Sem_2204_the_check_operation_083 | Verify behaviour of `port.check(catch)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 89 | Sem_2204_the_check_operation_084 | Verify behaviour of `port.check(catch)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 90 | Sem_2204_the_check_operation_085 | Verify behaviour of any `port.check(catch)` in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 91 | Sem_2204_the_check_operation_086 | Verify behaviour of any `port.check(catch)` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 92 | Sem_2204_the_check_operation_087 | Verify behaviour of any `port.check(catch)` in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 93 | Sem_2204_the_check_operation_088 | Verify behaviour of any `port.check(catch)` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | n |
| 94 | Sem_2204_the_check_operation_089 | Verify behaviour of `port.check(catch)` in case of successful match inside `alt` | Clause 22.4 | m | y |
| 95 | Sem_2204_the_check_operation_090 | Verify behaviour of `port.check(catch)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | y |
| 96 | Sem_2204_the_check_operation_091 | Verify behaviour of `port.check(catch)` in case of successful match in standalone statement | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 97 | Sem_2204_the_check_operation_092 | Verify behaviour of `port.check(catch)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | y |
| 98 | Sem_2204_the_check_operation_093 | Verify behaviour of any `port.check(catch)` in case of successful match inside `alt` | Clause 22.4 | m | n |
| 99 | Sem_2204_the_check_operation_094 | Verify behaviour of any `port.check(catch)` with assignment in case of successful match inside `alt` | Clause 22.4 | m | n |
| 100 | Sem_2204_the_check_operation_095 | Verify behaviour of any `port.check(catch)` in case of successful match in standalone statement | Clause 22.4 | m | n |
| 101 | Sem_2204_the_check_operation_096 | Verify behaviour of any `port.check(catch)` with assignment in case of successful match in standalone statement | Clause 22.4 | m | n |
| 102 | Sem_2204_the_check_operation_097 | Verify that `port.check` works correctly inside `alt` | Clause 22.4 | m | y |
| 103 | Sem_2204_the_check_operation_098 | Verify that `port.check` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 104 | Sem_2204_the_check_operation_099 | Verify that `port.check` works correctly as standalone statement | Clause 22.4 | m | y |
| 105 | Sem_2204_the_check_operation_100 | Verify that `port.check` with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 106 | Sem_2204_the_check_operation_101 | Verify that any `port.check` works correctly inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 107 | Sem_2204_the_check_operation_102 | Verify that any `port.check` with assignment works correctly inside `alt` | Clause 22.4 | m | y |
| 108 | Sem_2204_the_check_operation_103 | Verify that any `port.check` works correctly as standalone statement | Clause 22.4 | m | y |
| 109 | Sem_2204_the_check_operation_104 | Verify that any `port.check`(catch) with assignment works correctly as standalone statement | Clause 22.4 | m | y |
| 110 | Sem_2204_the_check_operation_105 | Verify behaviour of `port.check` in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 111 | Sem_2204_the_check_operation_106 | Verify behaviour of `port.check` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |
| 112 | Sem_2204_the_check_operation_107 | Verify behaviour of `port.check` in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 113 | Sem_2204_the_check_operation_108 | Verify behaviour of `port.check` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 114 | Sem_2204_the_check_operation_109 | Verify any `port.check` behaviour in case of unsuccessful match inside `alt` | Clause 22.4 | m | n |
| 115 | Sem_2204_the_check_operation_110 | Verify behaviour of any `port.check` with assignment in case of unsuccessful match inside `alt` | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 116 | Sem_2204_the_check_operation_111 | Verify behaviour of any `port.check` in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 117 | Sem_2204_the_check_operation_112 | Verify behaviour of any `port.check` with assignment in case of unsuccessful match in standalone statement | Clause 22.4 | m | y |
| 118 | Sem_2204_the_check_operation_113 | Verify behaviour of `port.check` in case of successful match inside `alt` | Clause 22.4 | m | n |
| 119 | Sem_2204_the_check_operation_114 | Verify behaviour of `port.check` with assignment in case of successful match inside `alt` | Clause 22.4 | m | y |
| 120 | Sem_2204_the_check_operation_115 | Verify behaviour of `port.check` in case of successful match in standalone statement | Clause 22.4 | m | y |
| 121 | Sem_2204_the_check_operation_116 | Verify behaviour of `port.check` with assignment in case of successful match in standalone statement | Clause 22.4 | m | y |
| 122 | Sem_2204_the_check_operation_117 | Verify behaviour of any `port.check` in case of successful match inside `alt` | Clause 22.4 | m | y |
| 123 | Sem_2204_the_check_operation_118 | Verify behaviour of any `port.check` with assignment in case of successful match inside `alt` | Clause 22.4 | m | y |
| 124 | Sem_2204_the_check_operation_119 | Verify behaviour of any `port.check` in case of successful match in standalone statement | Clause 22.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 125 | Sem_2204_the_check_operation_120 | Verify behaviour of any `port.check` with assignment in case of successful match in standalone statement | Clause 22.4 | m | y |

## 5.144. `Timer` operations

*Table 143. `Timer` operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_23_toplevel_001 | Ensure `timer` operations are not allowed outside of module control, test case, function, altstep | Clause 23 | m | y |
| 2 | NegSem_23_toplevel_002 | Ensure `timer` operations are not allowed outside of module control, test case, function, altstep | Clause 23 | m | y |
| 3 | NegSyn_23_toplevel_001 | Ensure `timer` operations are not allowed outside of module control, test case, function, altstep | Clause 23 | m | y |
| 4 | NegSyn_23_toplevel_002 | Ensure `timer` operations are not allowed outside of module control, test case, function, altstep | Clause 23 | m | y |
| 5 | Syn_23_toplevel_001 | Ensure timer allowed in module control, test case, function, altstep | Clause 23 | m | y |
| 6 | Syn_23_toplevel_002 | Ensure timer allowed in module control, test case, function, altstep | Clause 23 | m | y |

## 5.145. The `start` timer operation

*Table 144. The `start` timer operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2302_timer_start_001 | Ensure infinity is not allowed | Clause 23.2 | m | y |
| 2 | NegSem_2302_timer_start_002 | Ensure not_a_number is not allowed | Clause 23.2 | m | y |
| 3 | NegSem_2302_timer_start_003 | Ensure negative value is not allowed | Clause 23.2 | m | y |
| 4 | NegSem_2302_timer_start_004 | Ensure negative infinity is not allowed | Clause 23.2 | m | y |
| 5 | NegSyn_2302_timer_start_001 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 6 | NegSyn_2302_timer_start_002 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 7 | NegSyn_2302_timer_start_003 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 8 | NegSyn_2302_timer_start_004 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 9 | NegSyn_2302_timer_start_005 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 10 | NegSyn_2302_timer_start_006 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 11 | NegSyn_2302_timer_start_007 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 12 | NegSyn_2302_timer_start_008 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 13 | NegSyn_2302_timer_start_009 | Ensure `timer start` syntax | Clause 23.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | NegSyn_2302_timer_start_010 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 15 | NegSyn_2302_timer_start_011 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 16 | NegSyn_2302_timer_start_012 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 17 | NegSyn_2302_timer_start_013 | Ensure `timer start` syntax | Clause 23.2 | m | y |
| 18 | Sem_2302_timer_start_001 | Ensure timer runs from zero to stated value | Clause 23.2 | m | y |
| 19 | Sem_2302_timer_start_002 | Ensure timer can be restarted | Clause 23.2 | m | y |
| 20 | Sem_2302_timer_start_003 | Ensure timer default value can be modified by start value | Clause 23.2 | m | y |
| 21 | Sem_2302_timer_start_004 | Ensure timer with value 0.0 expires immediately | Clause 23.2 | m | y |

# 5.146. The `stop` timer operation

*Table 145. The `stop` timer operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_2303_timer_stop_001 | Ensure `timer stop` syntax | Clause 23.3 | m | y |
| 2 | NegSyn_2303_timer_stop_002 | Ensure `timer stop` syntax | Clause 23.3 | m | y |
| 3 | NegSyn_2303_timer_stop_003 | Ensure all `timer stop` syntax | Clause 23.3 | m | y |
| 4 | NegSyn_2303_timer_stop_004 | Ensure all `timer stop` syntax | Clause 23.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSyn_2303_timer_stop_005 | Ensure all `timer stop` syntax | Clause 23.3 | m | y |
| 6 | NegSyn_2303_timer_stop_006 | Ensure all `timer stop` syntax | Clause 23.3 | m | y |
| 7 | Sem_2303_timer_stop_002 | Ensure timer stop sets elapsed time to zero | Clause 23.3 | m | y |
| 8 | Sem_2303_timer_stop_003 | Ensure timer all timer identifier | Clause 23.3 | m | y |
| 9 | Sem_2303_timer_stop_004 | Ensure can be stopped after timeout | Clause 23.3 | m | y |
| 10 | Syn_2303_timer_stop_006 | Ensure `timer stop` syntax | Clause 23.3 | m | y |
| 11 | Syn_2303_timer_stop_007 | Ensure all `timer stop` syntax | Clause 23.3 | m | y |

# 5.147. The `read` timer operation

*Table 146. The `read` timer operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_2304_timer_read_001 | Ensure `timer read` syntax | Clause 23.4 | m | y |
| 2 | NegSyn_2304_timer_read_002 | Ensure `timer read` syntax | Clause 23.4 | m | y |
| 3 | NegSyn_2304_timer_read_003 | Ensure `timer read` syntax | Clause 23.4 | m | y |
| 4 | NegSyn_2304_timer_read_004 | Ensure `timer read` syntax: disallow any `timer.read` | Clause 23.4 | m | y |
| 5 | NegSyn_2304_timer_read_005 | Ensure `timer read` syntax | Clause 23.4 | m | y |
| 6 | Sem_2304_timer_read_001 | Ensure `timer read` result of inactive timer is zero | Clause 23.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 7 | Sem_2304_timer_read_002 | Ensure `timer read` result is non-negative float | Clause 23.4 | m | y |
| 8 | Sem_2304_timer_read_003 | Ensure `timer read` result is non-negative float | Clause 23.4 | m | y |

## 5.148. The `running` timer operation

*Table 147. The `running` timer operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_2305_timer_running_001 | Ensure `timer running` syntax | Clause 23.5 | m | y |
| 2 | NegSyn_2305_timer_running_002 | Ensure `timer running` syntax | Clause 23.5 | m | y |
| 3 | NegSyn_2305_timer_running_003 | Ensure `timer running` syntax | Clause 23.5 | m | y |
| 4 | NegSyn_2305_timer_running_004 | Ensure `timer running` syntax | Clause 23.5 | m | y |
| 5 | NegSyn_2305_timer_running_005 | Ensure `timer running` syntax | Clause 23.5 | m | y |
| 6 | NegSyn_2305_timer_running_006 | Ensure `timer running` syntax: disallow all `timer.running` | Clause 23.5 | m | y |
| 7 | Sem_2305_timer_running_001 | Ensure timer running any timer identifier works | Clause 23.5 | m | y |
| 8 | Sem_2305_timer_running_002 | Ensure `timer running` operation works | Clause 23.5 | m | y |
| 9 | Sem_2305_timer_running_003 | Ensure `timer running` operation works | Clause 23.5 | m | y |
| 10 | Sem_2305_timer_running_004 | Ensure `timer running` operation works | Clause 23.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_2305_timer_running_005 | Correct number of timers from a timer array is still running | Clause 23.5 | m | y |
| 12 | Syn_2305_timer_running_001 | Ensure `timer runnig` syntax | Clause 23.5 | m | y |

## 5.149. The `timeout` operation

*Table 148. The `timeout` operation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_2306_timer_timeout_001 | Ensure `timer timeout` syntax | Clause 23.6 | m | y |
| 2 | NegSyn_2306_timer_timeout_002 | Ensure `timer timeout` can`t be used in boolean expressions | Clause 23.6 | m | y |
| 3 | NegSyn_2306_timer_timeout_003 | Ensure `timer timeout` syntax | Clause 23.6 | m | y |
| 4 | NegSyn_2306_timer_timeout_004 | Ensure `timer timeout` syntax | Clause 23.6 | m | y |
| 5 | NegSyn_2306_timer_timeout_005 | Ensure `timer timeout` syntax | Clause 23.6 | m | y |
| 6 | NegSyn_2306_timer_timeout_006 | Ensure `timer timeout` syntax | Clause 23.6 | m | y |
| 7 | NegSyn_2306_timer_timeout_007 | Ensure` timer timeout` syntax | Clause 23.6 | m | y |
| 8 | Sem_2306_timer_timeout_001 | Ensure `timer timeout` operations: non-started timer does not timeout | Clause 23.6 | m | y |
| 9 | Sem_2306_timer_timeout_002 | Ensure `timer timeout` operations: timed-out timer does not timeout until restarted | Clause 23.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 10 | Sem_2306_timer_timeout_003 | Ensure `timer timeout` happen in order from the shortest to the longest | Clause 23.6 | m | y |
| 11 | Sem_2306_timer_timeout_004 | Ensure any `timer.timeout` operation | Clause 23.6 | m | y |
| 12 | Sem_2306_timer_timeout_005 | Ensure any `timer.timeout` operation for timeouts that are not in scope | Clause 23.6 | m | y |
| 13 | Sem_2306_timer_timeout_006 | Ensure any `timer.timeout` operation handles timeout of any timer in the component, not only visible from a function or altstep | Clause 23.6 | m | y |
| 14 | Sem_2306_timer_timeout_007 | Ensure `timer timeout` happen in order from the shortest to the longest | Clause 23.6 | m | y |
| 15 | Sem_2306_timer_timeout_008 | Timeout of a timer from a timer array works correctly | Clause 23.6 | m | y |
| 16 | Sem_2306_timer_timeout_009 | removing random timeout when using any `timer.timeout` | Clause 23.6 | m | y |

# 5.150. Test `verdict` operations

*Table 149. Test `verdict` operations*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_24_toplevel_001 | Ensure `getverdict` is not allowed in constant initialization in control part | Clause 24 | m | y |
| 2 | NegSem_24_toplevel_002 | Ensure `getverdict` is not allowed in parameter initialization in control part. | Clause 24 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_24_toplevel_003 | Ensure `getverdict` is not allowed in variable definition in control part. | Clause 24 | m | y |
| 4 | NegSem_24_toplevel_004 | Ensure `setverdict` is not allowed in part whithin `compound` statement. | Clause 24 | m | y |
| 5 | NegSem_24_toplevel_005 | Ensure `setverdict` is not allowed in control part at the top level. | Clause 24 | m | y |
| 6 | Syn_24_toplevel_001 | Ensure `setverdict` and `getverdict` are allowed in functions | Clause 24 | m | n |
| 7 | Syn_24_toplevel_002 | Ensure `setverdict` and `getverdict` are allowed in test cases | Clause 24 | m | n |
| 8 | Syn_24_toplevel_003 | Ensure `setverdict` and `getverdict` are allowed in altsteps | Clause 24 | m | y |

# 5.151. The verdict mechanism

*Table 150. The verdict mechanism*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2401_SetverdictError | `Setverdict` can't set error `verdict` | Clause 24.1 | m | y |
| 2 | Sem_2401_GlobalVerdict_001 | Ensure overwriting rules for global verdict: `pass` can overwrite `none`. | Clause 24.1 | m | y |
| 3 | Sem_2401_GlobalVerdict_002 | Ensure overwriting rules for global verdict: `inconc` can overwrite `none`. | Clause 24.1 | m | y |
| 4 | Sem_2401_GlobalVerdict_003 | Ensure overwriting rules for global verdict: `fail` can overwrite `none`. | Clause 24.1 | m | y |
| 5 | Sem_2401_GlobalVerdict_004 | Ensure overwriting rules for global verdict: `none` can't overwrite `pass`. | Clause 24.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_2401_GlobalVerdict_005 | Ensure overwriting rules for global verdict: `inconc` can overwrite `pass`. | Clause 24.1 | m | y |
| 7 | Sem_2401_GlobalVerdict_006 | Ensure overwriting rules for global verdict: `fail` can overwrite `pass`. | Clause 24.1 | m | y |
| 8 | Sem_2401_GlobalVerdict_007 | Ensure overwriting rules for global verdict: `none` can't overwrite `inconc`. | Clause 24.1 | m | y |
| 9 | Sem_2401_GlobalVerdict_008 | Ensure overwriting rules for global verdict: `pass` can't overwrite `inconc`. | Clause 24.1 | m | y |
| 10 | Sem_2401_GlobalVerdict_009 | Ensure overwriting rules for global verdict: `fail` can overwrite `inconc`. | Clause 24.1 | m | y |
| 11 | Sem_2401_GlobalVerdict_010 | Ensure overwriting rules for global verdict: `none` can't overwrite `fail`. | Clause 24.1 | m | y |
| 12 | Sem_2401_GlobalVerdict_011 | Ensure overwriting rules for global verdict: `pass` can't overwrite `fail`. | Clause 24.1 | m | y |
| 13 | Sem_2401_GlobalVerdict_012 | Ensure overwriting rules for global verdict: `inconc` can't overwrite `fail`. | Clause 24.1 | m | y |
| 14 | Sem_2401_InitiallyNone_001 | Local verdicts initializes with `none` | Clause 24.1 | m | y |
| 15 | Sem_2401_LocalVerdict_001 | Ensure overwriting rules for local verdict: `pass` can overwrite `none`. | Clause 24.1 | m | y |
| 16 | Sem_2401_LocalVerdict_002 | Ensure overwriting rules for local verdict: `inconc` can overwrite `none`. | Clause 24.1 | m | y |
| 17 | Sem_2401_LocalVerdict_003 | Ensure overwriting rules for local verdict: `fail` can overwrite `none`. | Clause 24.1 | m | y |
| 18 | Sem_2401_LocalVerdict_004 | Ensure overwriting rules for local verdict: `none` can't overwrite `pass`. | Clause 24.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 19 | Sem_2401_LocalVerdict_005 | Ensure overwriting rules for local verdict: `inconc` can overwrite `pass`. | Clause 24.1 | m | y |
| 20 | Sem_2401_LocalVerdict_006 | Ensure overwriting rules for local verdict: `fail` can overwrite `pass`. | Clause 24.1 | m | y |
| 21 | Sem_2401_LocalVerdict_007 | Ensure overwriting rules for local verdict: `none` can't overwrite `inconc`. | Clause 24.1 | m | y |
| 22 | Sem_2401_LocalVerdict_008 | Ensure overwriting rules for local verdict: `pass` can't overwrite `inconc`. | Clause 24.1 | m | y |
| 23 | Sem_2401_LocalVerdict_009 | Ensure overwriting rules for local verdict: `fail` can overwrite `inconc`. | Clause 24.1 | m | y |
| 24 | Sem_2401_LocalVerdict_010 | Ensure overwriting rules for local verdict: `none` can't overwrite `fail`. | Clause 24.1 | m | y |
| 25 | Sem_2401_LocalVerdict_011 | Ensure overwriting rules for local verdict: `pass` can't overwrite `fail`. | Clause 24.1 | m | y |
| 26 | Sem_2401_LocalVerdict_012 | Ensure overwriting rules for local verdict: inconc can't overwrite `fail`. | Clause 24.1 | m | y |
| 27 | Syn_2401_FiveValues_001 | There are five values of `verdicttype` | Clause 24.1 | m | y |

# 5.152. The setverdict mechanism

*Table 151. Test setverdict mechanism*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2402_setverdict_params_001 | Ensure `setverdict` accepts parameters of `verdicttype` only | Clause 24.2 | m | y |
| 2 | NegSem_2402_setverdict_params_002 | Ensure `setverdict` accepts parameters of `verdicttype` only | Clause 24.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | NegSem_2402_setverdict_params_003 | Ensure `setverdict` accepts values of `verdicttype` only | Clause 24.2 | m | y |
| 4 | NegSem_2402_setverdict_params_004 | Ensure `setverdict` accepts values only as the parameter | Clause 24.2 | m | y |
| 5 | NegSem_2402_setverdict_params_005 | Ensure `setverdict` accepts values only as the parameter | Clause 24.2 | m | y |
| 6 | Sem_2402_setverdict_logging_001 | Ensure logging constraints | Clause 24.2 | m | y |
| 7 | Sem_2402_setverdict_params_001 | Ensure `setverdict` accepts values only as the parameter | Clause 24.2 | m | y |
| 8 | Sem_2402_setverdict_params_002 | Ensure `setverdict` accepts values only as the parameter | Clause 24.2 | m | y |
| 9 | Sem_2402_setverdict_params_003 | Ensure logging constraints | Clause 24.2 | m | n |

# 5.153. The getverdict mechanism

*Table 152. The getverdict mechanism*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_2403_getverdict_001 | Ensure `getverdict` returns the actual verdict `none` | Clause 24.3 | m | y |
| 2 | Sem_2403_getverdict_002 | Ensure `getverdict` returns the actual verdict `inconc` | Clause 24.3 | m | y |
| 3 | Sem_2403_getverdict_003 | Ensure `getverdict` returns the actual verdict `pass` | Clause 24.3 | m | y |
| 4 | Sem_2403_getverdict_004 | Ensure `getverdict` returns the actual verdict `fail` | Clause 24.3 | m | y |
| 5 | Sem_2403_getverdict_005 | Ensure `getverdict` `none` for uninitialized verdict | Clause 24.3 | m | y |

# 5.154. Module control

*Table 153. Module control*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_26_ModuleControl_001 | Assignments in the control part are accepted. | Clause 26 | m | y |
| 2 | Syn_26_ModuleControl_002 | `if-else` constructs in the control part are accepted. | Clause 26 | m | y |
| 3 | Syn_26_ModuleControl_003 | `select-case` constructs in the control part are accepted. | Clause 26 | m | y |
| 4 | Syn_26_ModuleControl_004 | `for loop` constructs in the control part are accepted. | Clause 26 | m | y |
| 5 | Syn_26_ModuleControl_005 | `while loop` constructs in the control part are accepted. | Clause 26 | m | y |
| 6 | Syn_26_ModuleControl_006 | `label` and `goto` constructs in the control part are accepted. | Clause 26 | m | y |
| 7 | Syn_26_ModuleControl_007 | The `stop` construct in the control part is accepted. | Clause 26 | m | y |
| 8 | Syn_26_ModuleControl_008 | The `break` construct in the control part is accepted. | Clause 26 | m | y |
| 9 | Syn_26_ModuleControl_009 | The `continue` construct in the control part is accepted. | Clause 26 | m | y |
| 10 | Syn_26_ModuleControl_010 | The `continue` construct in the control part is accepted. | Clause 26 | m | y |
| 11 | Syn_26_ModuleControl_011 | The `alt/timeout` construct in the control part is accepted. | Clause 26 | m | y |
| 12 | Syn_26_ModuleControl_012 | The `repeat` construct in the control part is accepted. | Clause 26 | m | y |
| 13 | Syn_26_ModuleControl_013 | The `interleave` construct in the control part is accepted. | Clause 26 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 14 | Syn_26_ModuleControl_015 | `start`/`stop`/`read`/`running` timer constructs in the control part are accepted. | Clause 26 | m | y |
| 15 | Syn_26_ModuleControl_016 | The `action` construct in the control part is accepted. | Clause 26 | m | y |
| 16 | Syn_26_ModuleControl_017 | The `execute` construct in the control part is accepted. | Clause 26 | m | y |

## 5.155. The `execute` statement

*Table 154. The `execute` statement*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2601_ExecuteStatement_001 | Non-float `timeout` parameters in the `execute` statement are rejected (in this case int). | Clause 26.1 | m | y |
| 2 | NegSem_2601_ExecuteStatement_002 | Non-float `timeout` parameters in the `execute` statement are rejected (in this case charstring). | Clause 26.1 | m | y |
| 3 | NegSem_2601_ExecuteStatement_003 | Host id can be only charstring. | Clause 26.1 | m | n |
| 4 | NegSem_2601_ExecuteStatement_004 | Execution rejects test case execution with infinity timer guard | Clause 26.1 | m | y |
| 5 | Sem_2601_ExecuteStatement_001 | Parameters are passed correctly into the test case. | Clause 26.1 | m | y |
| 6 | Sem_2601_ExecuteStatement_002 | Multiple parameters of different types are passed correctly into the test case. | Clause 26.1 | m | y |
| 7 | Sem_2601_ExecuteStatement_003 | The timeout specified with the `execute` statement is respected. | Clause 26.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 8 | Sem_2601_ExecuteStatement_004 | The verdict `none` works correctly. | Clause 26.1 | m | y |
| 9 | Sem_2601_ExecuteStatement_005 | The verdict `pass` works correctly. | Clause 26.1 | m | y |
| 10 | Sem_2601_ExecuteStatement_006 | The verdict `inconc` works correctly. | Clause 26.1 | m | y |
| 11 | Sem_2601_ExecuteStatement_007 | The timeout specified with the `execute` statement is respected. | Clause 26.1 | m | n |
| 12 | Sem_2601_ExecuteStatement_008 | The user error sets the verdict error correctly. | Clause 26.1 | m | y |
| 13 | Sem_2601_ExecuteStatement_009 | Host id restriction is correctly handled. | Clause 26.1 | m | n |
| 14 | Sem_2601_ExecuteStatement_010 | verify that test cases can be executed from altsteps called from the control block | Clause 26.1 | m | y |

# 5.156. The control part

*Table 155. The control part*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2602_TheControlPart_001 | `setverdict` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 2 | NegSem_2602_TheControlPart_002 | The `create` component is not allowed in the control part. | Clause 26.2 | m | y |
| 3 | NegSem_2602_TheControlPart_003 | The `create alive` component is not allowed in the control part. | Clause 26.2 | m | y |
| 4 | NegSem_2602_TheControlPart_004 | The `start` statement is not allowed in the control part. | Clause 26.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 5 | NegSem_2602_TheControlPart_005 | The `stop` statement is not allowed in the control part. | Clause 26.2 | m | y |
| 6 | NegSem_2602_TheControlPart_006 | The `kill` statement is not allowed in the control part. | Clause 26.2 | m | y |
| 7 | NegSem_2602_TheControlPart_007 | The `alive` operation is not allowed in the control part. | Clause 26.2 | m | y |
| 8 | NegSem_2602_TheControlPart_008 | The `running` operation is not allowed in the control part. | Clause 26.2 | m | y |
| 9 | NegSem_2602_TheControlPart_009 | The `done` operation is not allowed in the control part. | Clause 26.2 | m | y |
| 10 | NegSem_2602_TheControlPart_010 | The `killed` operation is not allowed in the control part. | Clause 26.2 | m | y |
| 11 | NegSem_2602_TheControlPart_011 | The `connect` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 12 | NegSem_2602_TheControlPart_012 | The `disconnect` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 13 | NegSem_2602_TheControlPart_013 | The `map` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 14 | NegSem_2602_TheControlPart_014 | The `unmap` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 15 | NegSem_2602_TheControlPart_015 | The `send` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 16 | NegSem_2602_TheControlPart_016 | The `receive` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 17 | NegSem_2602_TheControlPart_017 | The `call` statements are not allowed in the control part. | Clause 26.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 18 | NegSem_2602_TheControlPart_018 | The `reply` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 19 | NegSem_2602_TheControlPart_019 | The `raise` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 20 | NegSem_2602_TheControlPart_020 | The `trigger` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 21 | NegSem_2602_TheControlPart_021 | The `getcall` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 22 | NegSem_2602_TheControlPart_022 | The `getreply` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 23 | NegSem_2602_TheControlPart_023 | The `catch` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 24 | NegSem_2602_TheControlPart_024 | The `check` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 25 | NegSem_2602_TheControlPart_025 | The `clear` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 26 | NegSem_2602_TheControlPart_026 | The `start` statements on ports are not allowed in the control part. | Clause 26.2 | m | y |
| 27 | NegSem_2602_TheControlPart_027 | The `stop` statements on ports are not allowed in the control part. | Clause 26.2 | m | y |
| 28 | NegSem_2602_TheControlPart_028 | The `halt` statements are not allowed in the control part. | Clause 26.2 | m | y |
| 29 | NegSem_2602_TheControlPart_029 | Alternative behaviours are only used to control timer behaviour in the control part. | Clause 26.2 | m | y |
| 30 | NegSem_2602_TheControlPart_030 | `Getverdict` statements are not allowed in the control part. | Clause 26.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 31 | NegSem_2602_TheControlPart_031 | `Execute` statements are not executed from test cases. | Clause 26.1 | m | y |
| 32 | NegSem_2602_TheControlPart_032 | The `create alive` named component is not allowed in the control part. | Clause 26.2 | m | y |
| 33 | NegSem_2602_TheControlPart_033 | The `create` named component is not allowed in the control part. | Clause 26.2 | m | y |
| 34 | NegSem_2602_TheControlPart_034 | The `create` named component on host is not allowed in the control part. | Clause 26.2 | m | y |
| 35 | NegSem_2602_TheControlPart_035 | Alternative behaviours are only used to control timer behaviour in the control part. | Clause 26.2 | m | y |
| 36 | Sem_2602_TheControlPart_001 | The selection/deselection of test cases using boolean conditions works as expected. | Clause 26.2 | m | y |
| 37 | Sem_2602_TheControlPart_002 | The execution of test cases works from within a function. | Clause 26.2 | m | y |
| 38 | Sem_2602_TheControlPart_003 | The selection of test cases can be achieved based on resulting verdict types. | Clause 26.2 | m | y |

# 5.157. Scope of attributes

*Table 156. Scope of attributes*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Syn_270101_ScopeOfAttributes_001 | Attributes for language elements are accepted. | Clause 27.1.1 | m | y |
| 2 | Syn_270101_ScopeOfAttributes_002 | Attributes for language elements are accepted. | Clause 27.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | Syn_270101_ScopeOfAttributes_003 | Attributes for individual fields are accepted. | Clause 27.1.1 | m | y |
| 4 | Syn_270101_ScopeOfAttributes_004 | Attributes for individual fields are accepted. | Clause 27.1.1 | m | y |

# 5.158. Optional attributes

*Table 157. Optional attributes*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_2707_OptionalAttributes_001 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | n |
| 2 | NegSem_2707_OptionalAttributes_002 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | n |
| 3 | NegSem_2707_OptionalAttributes_003 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | n |
| 4 | Sem_2707_OptionalAttributes_001 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 5 | Sem_2707_OptionalAttributes_002 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 6 | Sem_2707_OptionalAttributes_003 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 7 | Sem_2707_OptionalAttributes_004 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 8 | Sem_2707_OptionalAttributes_005 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 9 | Sem_2707_OptionalAttributes_006 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 10 | Sem_2707_OptionalAttributes_007 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |
| 11 | Sem_2707_OptionalAttributes_008 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | n |
| 12 | Syn_2707_OptionalAttributes_001 | The IUT correctly handles attribute definitions and their scoping rules | Clause 27.7 | m | y |

# 5.159. Matching specific values

*Table 158. Matching specific values*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B0101_matching_specific_value_001 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 2 | Sem_B0101_matching_specific_value_002 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 3 | Sem_B0101_matching_specific_value_003 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 4 | Sem_B0101_matching_specific_value_004 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 5 | Sem_B0101_matching_specific_value_005 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 6 | Sem_B0101_matching_specific_value_006 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 7 | Sem_B0101_matching_specific_value_007 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 8 | Sem_B0101_matching_specific_value_008 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 9 | Sem_B0101_matching_specific_value_009 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 10 | Sem_B0101_matching_specific_value_010 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 11 | Sem_B0101_matching_specific_value_011 | The IUT correctly handles template matching of specific values | Clause B.1.1 | m | y |
| 12 | NegSem_B010101_omitting_values_001 | Ensure that the IUT correctly handles template matching of omitted values | Clause B.1.1 | m | y |
| 13 | Sem_B010101_omitting_values_001 | Ensure that the IUT correctly handles template matching of omitted values | Clause B.1.1 | m | y |
| 14 | Sem_B010101_omitting_values_002 | Ensure that the IUT correctly handles template matching of omitted values | Clause B.1.1 | m | y |

# 5.160. Value list

*Table 159. Value list*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010201_value_list_001 | The IUT correctly handles template matching of listed multiple values | Clause B.1.2.1 | m | y |
| 3 | NegSem_B010201_value_list_002 | The IUT correctly handles template matching of listed multiple values | Clause B.1.2.1 | m | y |
| 4 | NegSem_B010201_value_list_003 | The IUT correctly handles template matching of listed multiple values | Clause B.1.2.1 | m | y |
| 5 | NegSem_B010201_value_list_004 | The IUT correctly handles template list corretly | Clause B.1.2.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_B010201_value_list_001 | The IUT correctly handles template matching of listed multiple values | Clause B.1.2.1 | m | y |
| 7 | Sem_B010201_value_list_002 | The IUT correctly handles template matching with all `from` clause | Clause B.1.2.1 | m | y |
| 8 | Sem_B010201_value_list_003 | The IUT correctly handles template matching of listed multiple values | Clause B.1.2.1 | m | y |
| 9 | Sem_B010201_value_list_004 | The IUT correctly handles template list corretly | Clause B.1.2.1 | m | y |

# 5.161. Complemented value list

*Table 160. Complemented value list*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010202_complemented_value_list_001 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | n |
| 2 | NegSem_B010202_complemented_value_list_002 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 3 | NegSem_B010202_complemented_value_list_003 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 4 | NegSem_B010202_complemented_value_list_004 | The IUT correctly handles template matching of complemented value omit | Clause B.1.2.2 | m | n |
| 5 | Sem_B010202_complemented_value_list_001 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_B010202_complemented_value_list_002 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 7 | Sem_B010202_complemented_value_list_003 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 8 | Sem_B010202_complemented_value_list_004 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 9 | Sem_B010202_complemented_value_list_005 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 10 | Sem_B010202_complemented_value_list_006 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 11 | Sem_B010202_complemented_value_list_007 | The IUT correctly handles template matching of complemented value listing | Clause B.1.2.2 | m | y |
| 12 | Sem_B010202_complemented_value_list_008 | The IUT correctly handles template matching of complemented value omit | Clause B.1.2.2 | m | y |

# 5.162. Any value

*Table 161. Any value*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010203_any_value_001 | The IUT correctly handles template matching of ? values | Clause B.1.2.3 | m | y |
| 2 | Sem_B010203_any_value_002 | The IUT correctly handles template matching of ? values | Clause B.1.2.3 | m | y |

# 5.163. Any value or none

*Table 162. Any value or none*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010204_any_value_or_none_001 | The IUT correctly handles template matching of * values | Clause B.1.2.4 | m | y |
| 2 | NegSem_B010204_any_value_or_none_002 | The IUT correctly handles template matching of * values | Clause B.1.2.4 | m | y |
| 3 | Sem_B010204_any_value_or_none_001 | The IUT correctly handles template matching of * values | Clause B.1.2.4 | m | y |
| 4 | Sem_B010204_any_value_or_none_002 | AnyValueOrNone can be assigned to top-level template | Clause B.1.2.4 | m | y |
| 5 | Sem_B010204_any_value_or_none_003 | AnyValueOrNone can be used for matching optional fields | Clause B.1.2.4 | m | y |
| 6 | Sem_B010204_any_value_or_none_004 | AnyValueOrNone cannot be used for matching non-optional value | Clause B.1.2.4 | m | y |
| 7 | Sem_B010204_any_value_or_none_005 | AnyValueOrNone cannot be used for matching compulsory fields | Clause B.1.2.4 | m | y |

# 5.164. Value range

*Table 163. Value range*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010205_value_range_001 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 2 | NegSem_B010205_value_range_002 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 3 | NegSem_B010205_value_range_003 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_B010205_value_range_001 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 5 | Sem_B010205_value_range_002 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 6 | Sem_B010205_value_range_003 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 7 | Sem_B010205_value_range_004 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 8 | Sem_B010205_value_range_005 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 9 | Sem_B010205_value_range_006 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 10 | Sem_B010205_value_range_007 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |
| 11 | Sem_B010205_value_range_008 | The IUT correctly handles template matching of value range definitions | Clause B.1.2.5 | m | y |

# 5.165. SuperSet

*Table 164. SuperSet*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010206_superset_001 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 2 | NegSem_B010206_superset_002 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 3 | NegSem_B010206_superset_003 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | n |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | NegSem_B010206_superset_004 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 5 | NegSem_B010206_superset_005 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 6 | NegSem_B010206_superset_006 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 7 | NegSem_B010206_superset_007 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 8 | NegSem_B010206_superset_008 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 9 | Sem_B010206_superset_001 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 10 | Sem_B010206_superset_002 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 11 | Sem_B010206_superset_003 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 12 | Sem_B010206_superset_004 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 13 | Sem_B010206_superset_005 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 14 | Sem_B010206_superset_006 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |
| 15 | Sem_B010206_superset_007 | The IUT correctly handles template matching of superset definitions | Clause B.1.2.6 | m | y |
| 16 | Sem_B010206_superset_008 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 17 | Sem_B010206_superset_009 | The IUT correctly handles template matching of superset definition | Clause B.1.2.6 | m | y |

# 5.166. SubSet

*Table 165. SubSet*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010207_subset_001 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 2 | NegSem_B010207_subset_002 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 3 | NegSem_B010207_subset_003 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | n |
| 4 | NegSem_B010207_subset_004 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 5 | NegSem_B010207_subset_005 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 6 | NegSem_B010207_subset_006 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 7 | NegSem_B010207_subset_007 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 8 | NegSem_B010207_subset_008 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 9 | Sem_B010207_subset_001 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 10 | Sem_B010207_subset_002 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 11 | Sem_B010207_subset_003 | The IUT correctly handles template matching of subset definitions | Clause B.1.2.7 | m | y |
| 12 | Sem_B010207_subset_004 | The IUT correctly handles template matching of subset definition | Clause B.1.2.7 | m | y |
| 13 | Sem_B010207_subset_005 | The IUT correctly handles template matching of subset definition | Clause B.1.2.7 | m | y |
| 14 | Sem_B010207_subset_006 | The IUT correctly handles template matching of subset definition | Clause B.1.2.7 | m | y |
| 15 | Sem_B010207_subset_007 | The IUT correctly handles template matching of subset definition | Clause B.1.2.7 | m | y |
| 16 | Sem_B010207_subset_008 | The IUT correctly handles template matching of subset definition | Clause B.1.2.7 | m | y |

# 5.167. Omitting optional fields

*Table 166. Omitting optional fields*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010208_omit_value_001 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |
| 2 | NegSem_B010208_omit_value_002 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |
| 3 | NegSem_B010208_omit_value_003 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | n |
| 4 | Sem_B010208_omit_value_001 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |
| 5 | Sem_B010208_omit_value_002 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_B010208_omit_value_003 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |
| 7 | Sem_B010208_omit_value_004 | The IUT correctly handles template matching of omit values | Clause B.1.2.8 | m | y |

# 5.168. Decoded content

*Table 167. Any element*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010209_decoded_content_001 | The IUT correctly handles content decoding | Clause B.1.2.9 | m | y |
| 2 | Sem_B010209_decoded_content_002 | The IUT correctly handles content decoding | Clause B.1.2.9 | m | y |
| 3 | Sem_B010209_decoded_content_003 | The IUT correctly handles content decoding | Clause B.1.2.9 | m | y |
| 4 | Sem_B010209_decoded_content_004 | The IUT correctly handles content decoding | Clause B.1.2.9 | m | y |
| 5 | Sem_B010209_decoded_content_005 | The IUT correctly handles content decoding | Clause B.1.2.9 | m | y |

# 5.169. Enumerated value list

*Table 168. Enumerated value list*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010210_enumerated_value_list_001 | The IUT correctly handles enum matching | Clause B.1.2.10 | m | n |

# 5.170. Any element

*Table 169. Any element*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010301_any_element_001 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 2 | Sem_B010301_any_element_002 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 3 | Sem_B010301_any_element_003 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 4 | Sem_B010301_any_element_004 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 5 | Sem_B010301_any_element_005 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 6 | Sem_B010301_any_element_006 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 7 | Sem_B010301_any_element_007 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |
| 8 | Sem_B010301_any_element_008 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.1 | m | y |

# 5.171. Any number of elements of no element

*Table 170. Any number of elements of no element*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010302_any_number_of_elements_or_none_001 | The IUT correctly handles template matching of * symbols in value elements | Clause B.1.3.2 | m | y |
| 2 | Sem_B010302_any_number_of_elements_or_none_002 | The IUT correctly handles template matching of * symbols in value elements | Clause B.1.3.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 3 | Sem_B010302_any_number_of_elements_or_none_003 | The IUT correctly handles template matching of * symbols in value elements | Clause B.1.3.2 | m | y |

# 5.172. Permutation

*Table 171. Permutation*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010303_permutation_001 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |
| 2 | NegSem_B010303_permutation_002 | All from operand can be a record of or set of only | Clause B.1.3.3 | m | y |
| 3 | NegSem_B010303_permutation_003 | Type restriction for permutation elements is applied | Clause B.1.3.3 | m | y |
| 4 | NegSem_B010303_permutation_004 | Type restriction for all `from` clause in permutation is applied | Clause B.1.3.3 | m | n |
| 5 | NegSem_B010303_permutation_005 | Verify restriction on individual members of all from operand in permutation | Clause B.1.3.3 | m | y |
| 6 | NegSem_B010303_permutation_006 | Verify restriction on individual members of all from operand in permutation | Clause B.1.3.3 | m | y |
| 7 | Sem_B010303_permutation_001 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |
| 8 | Sem_B010303_permutation_002 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |
| 9 | Sem_B010303_permutation_003 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 10 | Sem_B010303_permutation_004 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |
| 11 | Sem_B010303_permutation_005 | The IUT correctly handles template matching of ? symbols in value elements | Clause B.1.3.3 | m | y |
| 12 | Sem_B010303_permutation_006 | The IUT correctly handles permutation within arrays | Clause B.1.3.3 | m | y |
| 13 | Sem_B010303_permutation_007 | All `from` clause can be used inside permutation | Clause B.1.3.3 | m | y |
| 14 | Sem_B010303_permutation_008 | All `from` clause operand can be a set of value | Clause B.1.3.3 | m | y |
| 15 | Sem_B010303_permutation_009 | All `from` clause operand can be a set of value | Clause B.1.3.3 | m | y |

# 5.173. Length restrictions

*Table 172. Length restrictions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010401_length_restrictions_001 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |
| 2 | NegSem_B010401_length_restrictions_002 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |
| 3 | Sem_B010401_length_restrictions_001 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |
| 4 | Sem_B010401_length_restrictions_002 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |
| 5 | Sem_B010401_length_restrictions_003 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 6 | Sem_B010401_length_restrictions_004 | The IUT correctly handles template matching of value length definitions | Clause B.1.4.1 | m | y |

## 5.174. The `ifpresent` indicator

*Table 173. The `ifpresent` indicator*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010402_ifPresent_indicator_001 | The IUT correctly handles template matching of `ifpresent` indicators | Clause B.1.4.2 | m | y |
| 2 | Sem_B010402_ifPresent_indicator_001 | The IUT correctly handles template matching of `ifpresent` indicators | Clause B.1.4.2 | m | y |
| 3 | Sem_B010402_ifPresent_indicator_002 | The IUT correctly handles template matching of `ifpresent` indicators | Clause B.1.4.2 | m | y |

## 5.175. Matching character pattern

*Table 174. Matching character pattern*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B0105_toplevel_001 | The IUT correctly handles template matching of character pattern definitions | Clause B.1.5 | m | y |
| 2 | Sem_B0105_toplevel_002 | The IUT correctly handles template quadruple and USI-like syntax matching of character pattern definitions | Clause B.1.5 | m | y |

## 5.176. Set expression

*Table 175. Set expression*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010501_set_expression_001 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 2 | Sem_B010501_set_expression_001 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 3 | Sem_B010501_set_expression_002 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 4 | Sem_B010501_set_expression_003 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 5 | Sem_B010501_set_expression_004 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 6 | Sem_B010501_set_expression_005 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |
| 7 | Sem_B010501_set_expression_006 | The IUT correctly handles template matching of character pattern set expressions | Clause B.1.5.1 | m | y |

# 5.177. Reference expression

*Table 176. Reference expression*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010502_reference_expression_001 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 2 | Sem_B010502_reference_expression_002 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 3 | Sem_B010502_reference_expression_003 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 4 | Sem_B010502_reference_expression_004 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 5 | Sem_B010502_reference_expression_005 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 6 | Sem_B010502_reference_expression_006 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 7 | Sem_B010502_reference_expression_007 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 8 | Sem_B010502_reference_expression_008 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 9 | Sem_B010502_reference_expression_009 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |
| 10 | Sem_B010502_reference_expression_010 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | n |
| 11 | Sem_B010502_reference_expression_011 | The IUT correctly handles template matching of character pattern reference expressions | Clause B.1.5.2 | m | y |

# 5.178. Match expression n times

*Table 177. Match expression n times*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010503_match_n_times_001 | The IUT correctly handles template matching of character pattern expression multiplicity | Clause B.1.5.3 | m | y |
| 2 | Sem_B010503_match_n_times_002 | The IUT correctly handles template matching of character pattern expression multiplicity | Clause B.1.5.3 | m | y |
| 3 | Sem_B010503_match_n_times_003 | The IUT correctly handles template matching of character pattern expression multiplicity | Clause B.1.5.3 | m | y |
| 4 | Sem_B010503_match_n_times_004 | The IUT correctly handles template matching of character pattern expression multiplicity | Clause B.1.5.3 | m | y |
| 5 | Sem_B010503_match_n_times_005 | The IUT correctly handles template matching of character pattern expression multiplicity | Clause B.1.5.3 | m | y |

# 5.179. Match a referenced character set

*Table 178. Match a referenced character set*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSem_B010504_match_referenced_characters_001 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |
| 2 | Sem_B010504_match_referenced_characters_001 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |
| 3 | Sem_B010504_match_referenced_characters_002 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_B010504_match_referenced_characters_003 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |
| 5 | Sem_B010504_match_referenced_characters_004 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | |
| 6 | Sem_B010504_match_referenced_characters_005 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |
| 7 | Sem_B010504_match_referenced_characters_006 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |
| 8 | Sem_B010504_match_referenced_characters_007 | The IUT correctly handles template matching of character pattern reference characters | Clause B.1.5.4 | m | y |

# 5.180. Type compatibility rules for patterns

*Table 179. Type compatibility rules for patterns*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | NegSyn_B010505_pattern_compatibility_001 | The IUT correctly handles character pattern metacharacters compatibility rules of template matching | Clause B.1.5.5 | m | y |
| 2 | Sem_B010505_pattern_compatibility_001 | The IUT correctly handles character pattern compatibility rules of template matching | Clause B.1.5.5 | m | y |
| 3 | Sem_B010505_pattern_compatibility_002 | The IUT correctly handles character pattern compatibility rules of template matching | Clause B.1.5.5 | m | n |

# 5.181. Case insensitive pattern matching

*Table 180. Case insensitive pattern matching*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_B010506_case_sensitive_pattern_matching_001 | The IUT correctly handles character pattern compatibility rules of template case sensitive matching (`@nocase`) | Clause B.1.5.6 | m | y |
| 2 | Sem_B010506_case_sensitive_pattern_matching_002 | The IUT correctly handles character pattern compatibility rules of template case sensitive matching (`@nocase`) | Clause B.1.5.6 | m | y |

# 5.182. Other functions

*Table 181. Other functions*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_C0602_The_hostid_function_001 | Ensure that the IUT correctly handles the hostid function | Clause C.6.3 | m | y |
| 2 | Sem_C0602_The_testcasename_function_001 | Ensure that the IUT correctly handles the testcasename function | Clause C.6.2 | m | y |

# 5.183. Preprocessing macros

*Table 182. Preprocessing macros*

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 1 | Sem_D01_macro_module_001 | **MODULE** replaces the module name | Clause D | m | y |
| 2 | Sem_D02_macro_file_001 | **FILE** macro stores the path and file name in a charstring | Clause D | m | y |
| 3 | Sem_D03_macro_bfile_001 | The **BFILE** macro replaces the actual file name | Clause D | m | y |

| Item | TC/TP reference | purpose | Reference in ES 201 873-1 | Status | Support |
|---|---|---|---|---|---|
| 4 | Sem_D04_macro_line_001 | **LINE** macro stores the actual line number when it is called | Clause D | m | y |
| 5 | NegSem_D05_macro_scope_001 | **SCOPE** replaces the actual higher named basic scope unit | Clause D | m | y |
| 6 | Sem_D05_macro_scope_001 | **SCOPE** replaces the actual higher basic unit | Clause D | m | y |
| 7 | Sem_D05_macro_scope_002 | **SCOPE** replaces the actual higher basic unit | Clause D | m | y |