ETSI TS 102 950-1 V1.3.1 (2013-10)

Methods for Testing and Specification (MTS);
TTCN-3 Conformance Test Suite;
Part 1: Implementation Conformance Statement (ICS)

Reference
RTS/MTS-102950-1ed131 T3Conf

Keywords
conformance, ICS, testing, TTCN

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The present document is part 1 of a multi-part deliverable covering a TTCN-3 conformance test suite, as identified below:

Part 1: "Implementation Conformance Statement";

Part 2: "Test Suite Structure and Test Purposes (TSS&TP)";

Part 3: "Abstract Test Suite (ATS) and Implementation eXtra Information for Testing (IXIT)".

1 Scope

The present document provides the Implementation Conformance Statement (ICS) proforma for the conformance test suite for TTCN-3 as defined in ES 201 873-1 [1] in compliance with the relevant guidance given in the proforma for TTCN-3 reference test suite TS 102 995 [4]. In the present document only the core language features, specified in ES 201 873-1 [1] have been considered but not the tool implementation (see [i.1] and [i.2]), language mapping (see [i.3], [i.4] and [i.5]) and language extension (see e.g. [i.6], [i.7] and [i.8]) aspects.

The supplier of an implementation which is claimed to conform to ES 201 873-1 [1] is required to complete a copy of the ICS proforma provided in the annex A of the present document.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

2.1 Normative references

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

[1] ETSI ES 201 873-1 (V4.3.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

[2] ISO/IEC 9646-7 (1995): "Information Technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 7: Implementation Conformance Statement".

[3] ISO/IEC 9646-1 (1994): "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.

[i.1] 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)".

[i.2] 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)".

[i.3] 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".

[i.4] 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".

[i.5] 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".

[i.6] 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".

[i.7] ETSI ES 202 784: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization".

[i.8] ETSI ES 202 785: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ISO/IEC 9646-1 [3], ISO/IEC 9646-7 [2], ES 201 873-1 [1] (TTCN-3) and the following apply:

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. The ICS can take several forms: protocol ICS, profile ICS, profile specific ICS, information object ICS, API ICS, etc.

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

For the purposes of the present document, the following abbreviations apply:

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	Testling and Test Control Notation
TTCN-3	Testing and Test Control Notation edition 3

4 Conformance to this ICS proforma specification

If it claims to conform to the present document, the actual ICS proforma to be filled in by a supplier shall be technically equivalent to the text of the ICS proforma given in annex A, and shall preserve the numbering/naming and ordering of the proforma items.

An ICS which conforms to the present document shall be a conforming ICS proforma completed in accordance with the instructions for completion given in clause A.1.

Annex A (normative): ICS proforma for TTCN-3 conformance

A.1 Guidance for completing the ICS proforma

A.1.1 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.

A.1.2 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 [1] may provide information about the implementation in a standardized manner.

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

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

A.1.3 Conventions

The ICS proforma is composed of information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7 [2].

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 [1], except where explicitly stated otherwise.

Status column

The following notations, defined in ISO/IEC 9646-7 [2], 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.

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 [1].

Support column

The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [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.

A.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.

A.2.1 Date of the statement

Date of the statement:	

A.2.2 Implementation under Test (IUT) identification

IUT name:	
IUT version:	

A.2.3 System under Test (SUT) identification

SUT name:	
Hardware configuration:	
Operating system:	

A.2.4 Product supplier

Name:	
Address:	
Telephone number:	
Facsimile number:	
E-mail address:	
Additional information:	

A.2.5 Client

Name:	
Address:	
Telephone number:	
Facsimile number:	
E-mail address:	
Additional information:	

A.2.6 ICS contact person

Name:	
Telephone number:	
Facsimile number:	
E-mail address:	
Additional information:	

A.3 ICS proforma tables

A.3.1 Global statement of conformance

	(Yes/No)
Are all mandatory capabilities implemented?	

NOTE: Answering "No" to this question indicates non-conformance to the TTCN-3 core language. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming.

A.3.2 Basic language elements

Table A.1: Basic language elements

Item	TC/TP reference	Purpose	Reference in ES 201 873-1 [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	

A.3.3 Identifiers and keywords

Table A.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	
2	NegSyn_0501_Identifier_001	Ensure that when the IUT loads a module containing an identifier named with a keyword then the module is rejected.	Clause 5.1	m	
3	Syn_0501_Identifier_001	The IUT handle the identifiers case sensitively.	Clause 5.1	m	

A.3.4 Scope rules

Table A.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	
2	NegSem_0502_Scope_002	The IUT correctly handles definitions of local scope.	Clause 5.2	m	
3	NegSem_0502_Scope_003	The IUT correctly handles definitions of local scope.	Clause 5.2	m	
4	Sem_0502_Scope_001	The IUT handle scope hiearchy of component constants.	Clause 5.2	m	
5	Sem_0502_Scope_002	The IUT handle scope hiearchy with component booleans.	Clause 5.2	m	
6	Sem_0502_Scope_003	The IUT handles scope hierarchy via functions.	Clause 5.2	m	
7	Sem_0502_Scope_004	The IUT correctly handles the scope of definitions made in the module part.	Clause 5.2	m	
8	Sem_0502_Scope_008	The IUT correctly handles definitions of extended component scope.	Clause 5.2	m	
9	Syn_0502_Scope_001	The IUT supports all the nine scope units.	Clause 5.2	m	

A.3.5 Scope of formal parameters

Table A.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	
2	Sem_050201_Scope_of_parameters_002	The IUT correctly handles scope of formal function parameters	Clause 5.2.1	m	

A.3.6 Uniqueness of identifiers

Table A.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	
2	NegSem_050202_Uniqueness_004	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
3	NegSem_050202_Uniqueness_005	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
4	NegSem_050202_Uniqueness_006	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
5	NegSem_050202_Uniqueness_007	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
6	NegSem_050202_Uniqueness_008	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
7	NegSem_050202_Uniqueness_009	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
8	NegSem_050202_Uniqueness_010	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
9	NegSem_050202_Uniqueness_011	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
10	NegSem_050202_Uniqueness_012	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
11	Sem_050202_Uniqueness_001	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
12	Sem_050202_Uniqueness_002	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	
13	Sem_050202_Uniqueness_003	The IUT correctly handles the uniqueness of variable names in its scope	Clause 5.2.2	m	

A.3.7 Ordering of language elements

Table A.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	
2	NegSem_0503_Ordering_002	Declarations are in the allowed ordering	Clause 5.3	m	
3	NegSem_0503_Ordering_003	Declarations are in the allowed ordering	Clause 5.3	m	
4	Sem_0503_Ordering_001	Allowed orderings of declarations are supported	Clause 5.3	m	
5	Sem_0503_Ordering_002	Allowed any ordering with component definitions are supported	Clause 5.3	m	
6	Sem_0503_Ordering_005	Allowed orderings of declarations are supported	Clause 5.3	m	

A.3.8 Parameterization

Table A.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	
2	NegSyn_0504_forbidden_parametrization_001	The IUT rejects forbidden module parametrization types.	Clause 5.4	m	
3	NegSyn_0504_forbidden_parametrization_002	The IUT rejects forbidden module parametrization types.	Clause 5.4	m	

A.3.9 Formal parameters of kind value

Table A.8: Formal parameters of kind value

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_05040101_parameters_of_kind_value_001	The IUT correctly handles parametrization through the use of module parameters.	Clause 5.4.1.1	m	
2	Sem_05040101_parameters_of_kind_value_002	The IUT correctly handles parametrization through the use of module parameters.	Clause 5.4.1.1	m	
3	Sem_05040101_parameters_of_kind_value_003	The IUT correctly handles parametrization through the use of module parameters.	Clause 5.4.1.1	m	
4	Sem_05040101_parameters_of_kind_value_004	The IUT correctly handles parametrization through the use of module parameters.	Clause 5.4.1.1	m	

A.3.10 Formal parameters of kind template

Table A.9: Formal parameters of kind template

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_05040102_parameters_of_kind_template_001	The IUT correctly handles parametrization through the use of parameterized templates.	Clause 5.4.1.2	m	
2	Sem_05040102_parameters_of_kind_template_002	The IUT correctly handles parametrization through the use of parameterized templates.	Clause 5.4.1.2	m	

A.3.11 Formal parameters of kind timer

Table A.10: Formal parameters of kind timer

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_05040103_parameters_of_kind_timer_001	The IUT correctly handles parametrization through the use of timer parameters.	Clause 5.4.1.3	m	

A.3.12 Formal parameters of kind port

Table A.11: Formal parameters of kind port

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_05040104_parameters_of_kind_port_001	The IUT accepts port parametrization types for functions.	Clause 5.4.1.4	m	

A.3.13 Actual parameters

Table A.12: Actual parameters

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_050402_actual_parameters_001	The IUT accepts allowed assignments of actual parameters.	Clause 5.4.2	m	
2	Sem_050402_actual_parameters_002	The IUT accepts nested assignment of actual parameters.	Clause 5.4.2	m	

A.3.14 Cyclic definitions

Table A.13: Cyclic definitions

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_0505_cyclic_definitions_001	The IUT correctly handles recursive functions	Clause 5.5	m	
2	Sem_0505_cyclic_definitions_002	The IUT correctly handles cyclic imports	Clause 5.5	m	

A.3.15 Simple basic types and values

Table A.14: 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	
2	NegSyn_060100_SimpleBasicTypes_002	Assign boolean to integer values	Clause 6.1.0	m	
3	NegSyn_060100_SimpleBasicTypes_003	Assign integer to float values	Clause 6.1.0	m	
4	NegSyn_060100_SimpleBasicTypes_004	Assign boolean to float values	Clause 6.1.0	m	
5	NegSyn_060100_SimpleBasicTypes_005	Assign verdicttype to float values	Clause 6.1.0	m	
6	NegSyn_060100_SimpleBasicTypes_006	Assign integer to verdicttype values	Clause 6.1.0	m	
7	Sem_060100_SimpleBasicTypes_001	Assign and read integer values	Clause 6.1.0	m	
8	Sem_060100_SimpleBasicTypes_002	Assign and read large integer values	Clause 6.1.0	m	
9	Sem_060100_SimpleBasicTypes_003	Assign and read float values	Clause 6.1.0	m	
10	Sem_060100_SimpleBasicTypes_004	Assign and read large float values	Clause 6.1.0	m	
11	Sem_060100_SimpleBasicTypes_005	Assign and read verdicts	Clause 6.1.0	m	
12	Syn_060100_SimpleBasicTypes_001	Assign different integer values	Clause 6.1.0	m	
13	Syn_060100_SimpleBasicTypes_002	Assign large integer values	Clause 6.1.0	m	
14	Syn_060100_SimpleBasicTypes_003	Assign different float values	Clause 6.1.0	m	
15	Syn_060100_SimpleBasicTypes_004	Assign small and large float values	Clause 6.1.0	m	
16	Syn_060100_SimpleBasicTypes_005	Accept float mantisa for float values	Clause 6.1.0	m	
17	Syn_060100_SimpleBasicTypes_006	Accept all verdict values	Clause 6.1.0	m	

A.3.16 Basic string types and values

Table A.15: 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	
2	NegSyn_060101_TopLevel_002	Assign string to bitstring values	Clause 6.1.1	m	
3	NegSyn_060101_TopLevel_003	Assign octetstring to bitstring values	Clause 6.1.1	m	
4	NegSyn_060101_TopLevel_004	Assign invalid hexstring value	Clause 6.1.1	m	
5	NegSyn_060101_TopLevel_005	Assign string to hexstring values	Clause 6.1.1	m	
6	NegSyn_060101_TopLevel_006	Assign octetstring to hexstring values	Clause 6.1.1	m	
7	NegSyn_060101_TopLevel_007	Assign invalid hexstring value	Clause 6.1.1	m	
8	NegSyn_060101_TopLevel_008	Assign string to octetstring values	Clause 6.1.1	m	
9	NegSyn_060101_TopLevel_009	Assign hexstring to octetstring values	Clause 6.1.1	m	
10	NegSyn_060101_TopLevel_010	Assign invalid hexstring value	Clause 6.1.1	m	
11	Sem_060101_TopLevel_001	Assign and read bitstring	Clause 6.1.1	m	
12	Sem_060101_TopLevel_002	Assign and read hexstring	Clause 6.1.1	m	
13	Sem_060101_TopLevel_003	Assign and read octetstring	Clause 6.1.1	m	
14	Sem_060101_TopLevel_004	Assign and read charstring	Clause 6.1.1	m	
15	Sem_060101_TopLevel_005	Assign and read universal charstring	Clause 6.1.1	m	
16	Sem_060101_TopLevel_006	Assign and read universal charstring	Clause 6.1.1	m	
17	Syn_060101_TopLevel_001	Assign different bitstring values	Clause 6.1.1	m	
18	Syn_060101_TopLevel_002	Assign different hexstring values	Clause 6.1.1	m	
19	Syn_060101_TopLevel_003	Assign different octetstring values	Clause 6.1.1	m	

A.3.17 Accessing individual string elements

Table A.16: Accessing individual string elements

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_06010101_AccessStringElements_001	Access bitstring elements	Clause 6.1.1.1	m	
2	Sem_06010101_AccessStringElements_002	Access octetstring elements	Clause 6.1.1.1	m	
3	Sem_06010101_AccessStringElements_003	Access hexstring elements	Clause 6.1.1.1	m	
4	Sem_06010101_AccessStringElements_004	Access bitstring elements	Clause 6.1.1.1	m	
5	Sem_06010101_AccessStringElements_005	Access hexstring elements	Clause 6.1.1.1	m	
6	Sem_06010101_AccessStringElements_006	Access octetstring elements	Clause 6.1.1.1	m	
7	Sem_06010101_AccessStringElements_007	Access charstring elements	Clause 6.1.1.1	m	
8	Sem_06010101_AccessStringElements_008	Access charstring elements	Clause 6.1.1.1	m	

A.3.18 Lists of values

Table A.17: 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	
2	NegSem_06010201_ListOfValues_002	Assign values to restricted hexstring.	Clause 6.1.2.1	m	
3	NegSem_06010201_ListOfValues_003	Assign values to restricted octetstring.	Clause 6.1.2.1	m	
4	NegSem_06010201_ListOfValues_004	Assign values to restricted charstring.	Clause 6.1.2.1	m	
5	NegSem_06010201_ListOfValues_005	Assign values to restricted integer.	Clause 6.1.2.1	m	
6	NegSem_06010201_ListOfValues_006	Assign values to restricted float.	Clause 6.1.2.1	m	
7	Sem_06010201_ListOfValues_001	Assign invalid values to restricted bitstring.	Clause 6.1.2.1	m	

A.3.19 Lists of types

Table A.18: 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	
2	Sem_06010202_ListOfTypes_001	Assign values to list of types restricted bitstring.	Clause 6.1.2.2	m	

A.3.20 Ranges

Table A.19: 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	
2	NegSem_06010203_Ranges_002	Assign invalid values to restricted integer.	Clause 6.1.2.3	m	
3	NegSem_06010203_Ranges_003	Assure that not_a_number is not allowed in float range subtyping.	Clause 6.1.2.3	m	
4	NegSem_06010203_Ranges_004	Assign invalid values to restricted integer with exclusive bounds.	Clause 6.1.2.3	m	
5	NegSem_06010203_Ranges_005	Assign invalid values to restricted integer with exclusive bounds.	Clause 6.1.2.3	m	
6	NegSem_06010203_Ranges_006	Assign range to boolean not permitted.	Clause 6.1.2.3	m	
7	NegSem_06010203_Ranges_007	Assign invalid value to range constrained charstring.	Clause 6.1.2.3	m	
8	NegSem_06010203_Ranges_008	Assign invalid value to range constrained charstring.	Clause 6.1.2.3	m	
9	NegSem_06010203_Ranges_009	Assign invalid value to range constrained charstring.	Clause 6.1.2.3	m	
10	NegSem_06010203_Ranges_010	Assign invalid values to restricted float.	Clause 6.1.2.3	m	
11	NegSem_06010203_Ranges_011	Assign invalid values to range restricted float.	Clause 6.1.2.3	m	
12	NegSem_06010203_Ranges_012	Assign invalid values to range excluded restricted float.	Clause 6.1.2.3	m	
13	NegSem_06010203_Ranges_013	Assign invalid value to range constrained universal charstring.	Clause 6.1.2.3	m	
14	NegSem_06010203_Ranges_014	Assign invalid value to range constrained universal charstring with mixed bounds.	Clause 6.1.2.3	m	
15	NegSem_06010203_Ranges_015	Assign invalid value to range constrained charstring.	Clause 6.1.2.3	m	
16	NegSem_06010203_Ranges_016	Invalid value infinity for range constrained charstring.	Clause 6.1.2.3	m	
17	NegSem_06010203_Ranges_017	Invalid value -infinity for range constrained charstring.	Clause 6.1.2.3	m	
18	Sem_06010203_Ranges_001	Assign values to range restricted integer.	Clause 6.1.2.3	m	
19	Sem_06010203_Ranges_002	Assign values to infinity range restricted integer.	Clause 6.1.2.3	m	
20	Sem_06010203_Ranges_003	Assign values to range restricted integer with exclusive bounds.	Clause 6.1.2.3	m	
21	Sem_06010203_Ranges_004	Assign values to range restricted cahrstring with inclusive bounds.	Clause 6.1.2.3	m	
22	Sem_06010203_Ranges_005	Assign values to range restricted cahrstring with exclusive bounds.	Clause 6.1.2.3	m	
23	Sem_06010203_Ranges_006	Assign values to range restricted cahrstring with mixed bounds.	Clause 6.1.2.3	m	
24	Sem_06010203_Ranges_007	Assign values to range restricted universal charstring.	Clause 6.1.2.3	m	
25	Sem_06010203_Ranges_008	Assign values to range restricted universal charstring with mixed bounds.	Clause 6.1.2.3	m	

A.3.21 String length restrictions

Table A.20: String length restrictions

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_06010204_StringLenghtRestrict_001	Assign invalid values to length restricted bitstring.	Clause 6.1.2.4	m	
2	NegSem_06010204_StringLenghtRestrict_002	Assign invalid values to length restricted bitstring.	Clause 6.1.2.4	m	
3	Sem_06010204_StringLenghtRestrict_001	Assign values to list of types restricted bitstring.	Clause 6.1.2.4	m	

A.3.22 Pattern subtyping of character string types

Table A.21: 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	
2	Sem_06010205_StringPattern_001	Assign values to pattern restricted character strings.	Clause 6.1.2.5	m	
3	Sem_06010205_StringPattern_002	Assign values to pattern restricted character strings.	Clause 6.1.2.5	m	

A.3.23 Mixing patterns, lists and ranges

Table A.22: 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	
2	NegSem_0601020601_MixingSubtype_002	Assign invalid values to mixed restricted integers.	Clause 6.1.2.6.1	m	
3	Sem_0601020601_MixingSubtype_001	Assign values to mixed restricted floats.	Clause 6.1.2.6.1	m	
4	Sem_0601020601_MixingSubtype_002	Assign values to mixed restricted integers.	Clause 6.1.2.6.1	m	

A.3.24 Using length restriction with other constraints

Table A.23: 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	
2	NegSem_0601020602_StringMixing_002	Assign invalid values to mixed restricted character strings.	Clause 6.1.2.6.2	m	
3	NegSem_0601020602_StringMixing_003	Assign invalid values to mixed restricted character strings.	Clause 6.1.2.6.2	m	
4	NegSem_0601020602_StringMixing_004	Assign invalid values to mixed restricted bit strings.	Clause 6.1.2.6.2	m	
5	NegSem_0601020602_StringMixing_005	Assign invalid values to mixed restricted hex strings.	Clause 6.1.2.6.2	m	
6	NegSem_0601020602_StringMixing_006	Assign invalid values to mixed restricted octet strings.	Clause 6.1.2.6.2	m	
7	Sem_0601020602_StringMixing_001	Assign values to mixed restricted character strings.	Clause 6.1.2.6.2	m	
8	Sem_0601020602_StringMixing_002	Assign values to mixed restricted character strings.	Clause 6.1.2.6.2	m	
9	Sem_0601020602_StringMixing_003	Assign values to mixed restricted character strings.	Clause 6.1.2.6.2	m	
10	Sem_0601020602_StringMixing_004	Assign values to mixed restricted bit strings.	Clause 6.1.2.6.2	m	
11	Sem_0601020602_StringMixing_005	Assign values to mixed restricted hex strings.	Clause 6.1.2.6.2	m	
12	Sem_0601020602_StringMixing_006	Assign values to mixed restricted octet strings.	Clause 6.1.2.6.2	m	

A.3.25 Structured types and values

Table A.24: Structured types and values

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_0602_TopLevel_001	Assignments with "implicit omit" attribute are correctly handled	Clause 6.2	m	
2	NegSem_0602_TopLevel_002	Assignments with "implicit omit" attribute are correctly handled	Clause 6.2	m	
3	NegSyn_0602_TopLevel_001	Invalid recursive union type definition causing an error	Clause 6.2	m	
4	NegSyn_0602_TopLevel_002	Invalid recursive record type definition causing an error	Clause 6.2	m	
5	NegSyn_0602_TopLevel_003	Combined value list and assignment notation not allowed in the same (immediate) context.	Clause 6.2	m	
6	NegSyn_0602_TopLevel_004	The omit keyword shall not be used for mandatory fields.	Clause 6.2	m	
7	NegSyn_0602_TopLevel_005	The omit keyword shall not be used for mandatory fields.	Clause 6.2	m	
8	Sem_0602_TopLevel_001	Assignments with "implicit omit" attribute are correctly handled	Clause 6.2	m	
9	Sem_0602_TopLevel_002	Assignments with "implicit omit" attribute are correctly handled	Clause 6.2	m	
10	Sem_0602_TopLevel_003	Assignments with "implicit omit" attribute are correctly handled	Clause 6.2	m	
11	Syn_0602_TopLevel_001	Valid recursive union type definition	Clause 6.2	m	
12	Syn_0602_TopLevel_002	Valid recursive record type definition	Clause 6.2	m	
13	Syn_0602_TopLevel_003	Valid recursive record type definition	Clause 6.2	m	
14	Syn_0602_TopLevel_004	constant definition of a record type.	Clause 6.2	m	
15	Syn_0602_TopLevel_005	Fields not mentioned are implicitly left unspecified.	Clause 6.2	m	

A.3.26 Record type and values

Table A.25: Record type and values

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_060201_RecordTypeValues_001	The dot notation used in record type definitions is correctly handled	Clause 6.2.1	m	
2	Sem_060201_RecordTypeValues_001	The dot notation used in record type definitions is correctly handled	Clause 6.2.1	m	
3	Sem_060201_RecordTypeValues_002	The dot notation used in record type definitions is correctly handled	Clause 6.2.1	m	
4	Sem_060201_RecordTypeValues_003	The dot notation used in record type definitions is correctly handled	Clause 6.2.1	m	
5	Sem_060201_RecordTypeValues_004	The dot notation used in record type definitions is correctly handled	Clause 6.2.1	m	
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	
7	Syn_060201_RecordTypeValues_002	The IUT correctly handles empty record definitions.	Clause 6.2.1	m	

A.3.27 Referencing elements of record of and set of types

Table A.26: 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	
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	
3	Sem_060203_records_and_sets_of_single_types_001	ensure that the inner type referencing is correctly handled	Clause 6.2.3.2	m	

A.3.28 Communication port types

Table A.27: 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 handled.	Clause 6.2.9	m	
2	NegSem_060209_CommunicationPortTypes_002	Restriction of port definitions are appropriately handled	Clause 6.2.9	m	
3	NegSem_060209_CommunicationPortTypes_003	Restriction of port definitions are appropriately handled	Clause 6.2.9	m	
4	Sem_060209_CommunicationPortTypes_001	Map param statements are allowed in testcase block.	Clause 6.2.9	m	
5	Sem_060209_CommunicationPortTypes_002	Unmap param statements are allowed in testcase block.	Clause 6.2.9	m	
6	Sem_060209_CommunicationPortTypes_003	Map and unmap param and local port address are allowed in a testcase block.	Clause 6.2.9	m	
7	Sem_060209_CommunicationPortTypes_004	Map and unmap param and local port address are allowed in a testcase block.	Clause 6.2.9	m	
8	Syn_060209_CommunicationPortTypes_001	Message-based ports are accepted.	Clause 6.2.9	m	
9	Syn_060209_CommunicationPortTypes_002	Message-based ports with address are accepted.	Clause 6.2.9	m	
10	Syn_060209_CommunicationPortTypes_003	Message-based ports are accepted.	Clause 6.2.9	m	
11	Syn_060209_CommunicationPortTypes_004	A address is allowed inside port definition	Clause 6.2.9	m	
12	Syn_060209_CommunicationPortTypes_005	Map param is accepted by the port definition.	Clause 6.2.9	m	
13	Syn_060209_CommunicationPortTypes_006	Unmap param is accepted by the port definition.	Clause 6.2.9	m	
14	Syn_060209_CommunicationPortTypes_007	Complex port definition are accepted.	Clause 6.2.9	m	

A.3.29 Addressing entities inside the SUT

Table A.28: 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	
2	Sem_060212_AddressingEntitiesInsideSut_001	Ensure null assignment is accepted for addresses	Clause 6.2.12	m	
3	Sem_060212_AddressingEntitiesInsideSut_002	The right port address is used	Clause 6.2.12	m	

A.3.30 Type compatibility of non-structured types

Table A.29: 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	
2	NegSem_060301_non_structured_types_002	The IUT correctly handles assignments from incompatible type ranges	Clause 6.3.1	m	
3	NegSem_060301_non_structured_types_003	The IUT correctly handles assignments from incompatible type ranges	Clause 6.3.1	m	
4	NegSem_060301_non_structured_types_004	The IUT correctly handles assignments from incompatible type ranges	Clause 6.3.1	m	
5	NegSem_060301_non_structured_types_005	The IUT correctly handles assignments from incompatible type ranges	Clause 6.3.1	m	
6	NegSem_060301_non_structured_types_006	The IUT correctly handles assignments from incompatible type ranges	Clause 6.3.1	m	
7	NegSem_060301_non_structured_types_007	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
8	NegSem_060301_non_structured_types_008	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
9	NegSem_060301_non_structured_types_009	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
10	NegSem_060301_non_structured_types_010	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
11	NegSem_060301_non_structured_types_011	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
12	NegSem_060301_non_structured_types_012	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
13	Sem_060301_non_structured_types_001	The IUT correctly handles assignments from compatible type ranges	Clause 6.3.1	m	
14	Sem_060301_non_structured_types_002	The IUT correctly handles assignments from compatible size restrictions	Clause 6.3.1	m	
15	Sem_060301_non_structured_types_003	The IUT correctly handles assignments from compatible type ranges	Clause 6.3.1	m	
16	Sem_060301_non_structured_types_004	The IUT correctly handles assignments from compatible type ranges	Clause 6.3.1	m	

A.3.31 Type compatibility of structured types

Table A.30: 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	
2	NegSem_060302_structured_types_003	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
3	NegSem_060302_structured_types_004	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
4	NegSem_060302_structured_types_005	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
5	NegSem_060302_structured_types_006	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
6	NegSem_060302_structured_types_007	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
7	NegSem_060302_structured_types_008	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
8	NegSem_060302_structured_types_009	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
9	NegSem_060302_structured_types_010	The IUT rejects assignments from incompatible types or type ranges	Clause 6.3.2	m	
10	NegSem_060302_structured_types_011	The IUT rejects assignments from structures having incompatible anytypes	Clause 6.3.2	m	
11	NegSem_060302_structured_types_012	The IUT rejects assignments having mismatch between undefined and omitted elements	Clause 6.3.2	m	
12	NegSem_060302_structured_types_013	The IUT rejects assignments having mismatch between undefined and omitted elements	Clause 6.3.2	m	
13	NegSem_060302_structured_types_014	The IUT rejects assignments between incompatible structures	Clause 6.3.2	m	
14	NegSem_060302_structured_types_015	The IUT rejects assignments between incompatible structures	Clause 6.3.2	m	
15	NegSem_060302_structured_types_016	The IUT rejects assignments between incompatible structures	Clause 6.3.2	m	
16	NegSem_060302_structured_types_017	The IUT rejects assignments between incompatible structures	Clause 6.3.2	m	
17	NegSem_060302_structured_types_018	The IUT rejects assignments between incompatible structures	Clause 6.3.2	m	
18	NegSem_060302_structured_types_019	The IUT correctly handles assignments from structures having compatible types and lengths	Clause 6.3.2	m	

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

A.3.32 Type compatibility of enumerated types

Table A.31: 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	

A.3.33 Type compatibility of component types

Table A.32: 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	
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	
3	Sem_060303_component_types_001	The IUT correctly handles assignments from structures having compatible components	Clause 6.3.3	m	
4	Sem_060303_component_types_002	The IUT correctly handles assignments from structures having compatible components	Clause 6.3.3	m	

A.3.34 Arithmetic operators

Table A.33: Arithmetic operators

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_070101_ArithmeticOperators_001	The addition of two integer variables is evaluated correctly.	Clause 7.1.1	m	
2	Sem_070101_ArithmeticOperators_002	The addition of multiple integer variables is evaluated correctly.	Clause 7.1.1	m	
3	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	
4	Sem_070101_ArithmeticOperators_004	The substraction of two integer variables is evaluated correctly.	Clause 7.1.1	m	
5	Sem_070101_ArithmeticOperators_005	The substraction of multiple integer variables is evaluated correctly.	Clause 7.1.1	m	
6	Sem_070101_ArithmeticOperators_006	The multiplication of two integer variables is evaluated correctly.	Clause 7.1.1	m	
7	Sem_070101_ArithmeticOperators_007	The multiplication of multiple integer variables is evaluated correctly.	Clause 7.1.1	m	
8	Sem_070101_ArithmeticOperators_008	The division of two integer variables is evaluated correctly.	Clause 7.1.1	m	
9	Sem_070101_ArithmeticOperators_009	The division of multiple integer variables is evaluated correctly.	Clause 7.1.1	m	
10	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	
11	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	
12	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	
13	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	
14	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	
15	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	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
16	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	
17	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	
18	Sem_070101_ArithmeticOperators_018	Operator combinations and the modulo operator on integer variables is evaluated correctly.	Clause 7.1.1	m	
19	Sem_070101_ArithmeticOperators_019	The addition operator works on float variables.	Clause 7.1.1	m	
20	Sem_070101_ArithmeticOperators_020	The substraction operator works on float variables.	Clause 7.1.1	m	
21	Sem_070101_ArithmeticOperators_021	The multiplication operator works on float variables.	Clause 7.1.1	m	
22	Sem_070101_ArithmeticOperators_022	The division operator works on float variables.	Clause 7.1.1	m	
23	Sem_070101_ArithmeticOperators_023	The combination of different operators works on float variables.	Clause 7.1.1	m	
24	Syn_070101_ArithmeticOperators_001	The addition of two integers in a constant is accepted.	Clause 7.1.1	m	
25	Syn_070101_ArithmeticOperators_002	The substraction of two integers in a constant is accepted.	Clause 7.1.1	m	
26	Syn_070101_ArithmeticOperators_003	The multiplication of two integers in a constant is accepted.	Clause 7.1.1	m	
27	Syn_070101_ArithmeticOperators_004	The division of two integers in a constant is accepted.	Clause 7.1.1	m	
28	Syn_070101_ArithmeticOperators_005	The modulo operator on two integers is accepted.	Clause 7.1.1	m	
29	Syn_070101_ArithmeticOperators_006	The remainder operator on two integers is accepted.	Clause 7.1.1	m	
30	Syn_070101_ArithmeticOperators_007	Operator combinations on integers is accepted.	Clause 7.1.1	m	
31	Syn_070101_ArithmeticOperators_008	The addition operator on float constants is accepted.	Clause 7.1.1	m	
32	Syn_070101_ArithmeticOperators_009	The substraction operator on float constants is accepted.	Clause 7.1.1	m	
33	Syn_070101_ArithmeticOperators_010	The multiplication operator on float constants is accepted.	Clause 7.1.1	m	
34	Syn_070101_ArithmeticOperators_011	The division operator on float constants is accepted.	Clause 7.1.1	m	
35	Syn_070101_ArithmeticOperators_012	A combination of operators on float constants is accepted.	Clause 7.1.1	m	

A.3.35 List operator

Table A.34: 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	
2	Sem_070102_ListOperator_002	The list operator on charstrings is evaluated correctly.	Clause 7.1.2	m	
3	Sem_070102_ListOperator_003	The list operator on record of is evaluated correctly.	Clause 7.1.2	m	
4	Sem_070102_ListOperator_004	The list operator on set of is evaluated correctly.	Clause 7.1.2	m	
5	Sem_070102_ListOperator_005	The list operator on arrays is evaluated correctly.	Clause 7.1.2	m	
6	Sem_070102_ListOperator_006	The list operator on record of is evaluated correctly.	Clause 7.1.2	m	

A.3.36 Relational operators

Table A.35: Relational operators

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_070103_RelationalOperators_001	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
2	Sem_070103_RelationalOperators_001	The equals operator on integers is evaluated correctly.	Clause 7.1.3	m	
3	Sem_070103_RelationalOperators_002	The equals operator on floats is evaluated correctly.	Clause 7.1.3	m	
4	Sem_070103_RelationalOperators_003	The equals operator on enumerations is evaluated correctly.	Clause 7.1.3	m	
5	Sem_070103_RelationalOperators_004	The less than operator on integers is evaluated correctly.	Clause 7.1.3	m	
6	Sem_070103_RelationalOperators_005	The less than operator on floats is evaluated correctly.	Clause 7.1.3	m	
7	Sem_070103_RelationalOperators_006	The less than operator on enumerations is evaluated correctly.	Clause 7.1.3	m	
8	Sem_070103_RelationalOperators_007	The less than or equal to operator on integers is evaluated correctly with differing values.	Clause 7.1.3	m	
9	Sem_070103_RelationalOperators_008	The less than or equal to operator on integers is evaluated correctly with equal values.	Clause 7.1.3	m	
10	Sem_070103_RelationalOperators_009	The less than or equal to operator on floats is evaluated correctly with differing values.	Clause 7.1.3	m	
11	Sem_070103_RelationalOperators_010	The less than or equal to operator on floats is evaluated correctly with equal values.	Clause 7.1.3	m	
12	Sem_070103_RelationalOperators_011	The less than or equal to operator on enumerations is evaluated correctly with differing values.	Clause 7.1.3	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
13	Sem_070103_RelationalOperators_012	The less than or equal to operator on enumerations is evaluated correctly with equal values.	Clause 7.1.3	m	
14	Sem_070103_RelationalOperators_013	The greater than operator on integers is evaluated correctly.	Clause 7.1.3	m	
15	Sem_070103_RelationalOperators_014	The less than operator on floats is evaluated correctly.	Clause 7.1.3	m	
16	Sem_070103_RelationalOperators_015	The less than operator on enumerations is evaluated correctly.	Clause 7.1.3	m	
17	Sem_070103_RelationalOperators_016	The greater than or equal to operator on integers is evaluated correctly with differing values.	Clause 7.1.3	m	
18	Sem_070103_RelationalOperators_017	The greater than or equal to operator on integers is evaluated correctly with equal values.	Clause 7.1.3	m	
19	Sem_070103_RelationalOperators_018	The greater than or equal to operator on floats is evaluated correctly with differing values.	Clause 7.1.3	m	
20	Sem_070103_RelationalOperators_019	The greater than or equal to operator on floats is evaluated correctly with equal values.	Clause 7.1.3	m	
21	Sem_070103_RelationalOperators_020	The less than or equal to operator on enumerations is evaluated correctly with differing values.	Clause 7.1.3	m	
22	Sem_070103_RelationalOperators_021	The greater than or equal to operator on enumerations is evaluated correctly with equal values.	Clause 7.1.3	m	
23	Sem_070103_RelationalOperators_022	The not equals operator on integers is evaluated correctly.	Clause 7.1.3	m	
24	Sem_070103_RelationalOperators_023	The not equals operator on floats is evaluated correctly.	Clause 7.1.3	m	
25	Sem_070103_RelationalOperators_024	The not equals operator on enumerations is evaluated correctly.	Clause 7.1.3	m	
26	Sem_070103_RelationalOperators_025	The equals operator on sets is evaluated correctly.	Clause 7.1.3	m	
27	Sem_070103_RelationalOperators_026	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
28	Sem_070103_RelationalOperators_027	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
29	Sem_070103_RelationalOperators_028	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
30	Sem_070103_RelationalOperators_029	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
31	Sem_070103_RelationalOperators_030	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
32	Sem_070103_RelationalOperators_031	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
33	Sem_070103_RelationalOperators_032	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
34	Sem_070103_RelationalOperators_033	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	
35	Sem_070103_RelationalOperators_034	The equals operator on records is evaluated correctly.	Clause 7.1.3	m	

A.3.37 Logical operators

Table A.36: 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	
2	Sem_070104_LogicalOperators_002	The the and operator with true and false as operands work on boolean variables.	Clause 7.1.4	m	

A.3.38 Bitwise operators

Table A.37: 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	
2	Sem_070105_BitwiseOperators_002	The bitwise negation operator works as expected on hexstrings.	Clause 7.1.5	m	

A.3.39 Shift operators

Table A.38: 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	
2	Sem_070106_ShiftOperators_002	The shift left operator works as expected on hexstrings.	Clause 7.1.6	m	
3	Sem_070106_ShiftOperators_003	The shift right operator works as expected on bitstrings.	Clause 7.1.6	m	
4	Sem_070106_ShiftOperators_004	The shift right operator works as expected on hexstrings.	Clause 7.1.6	m	

A.3.40 Rotate operators

Table A.39: 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	
2	Sem_070107_RotateOperators_002	The rotate left operator works as expected on hexstrings.	Clause 7.1.7	m	
3	Sem_070107_RotateOperators_003	The rotate right operator works as expected on bitstrings.	Clause 7.1.7	m	
4	Sem_070107_RotateOperators_004	The rotate right operator works as expected on hexstrings.	Clause 7.1.7	m	

A.3.41 Field references and list elements

Table A.40: 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	
2	Sem_0702_FieldReferencesAndListElements_002	The IUT correctly handles field referencing	Clause 7.2	m	

A.3.42 Definition of a module

Table A.41: 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	
2	Syn_0801_DefinitionOfAModule_001	A "plain" module definition is accepted.	Clause 8.1	m	
3	Syn_0801_DefinitionOfAModule_002	A module definition with language specification is accepted.	Clause 8.1	m	
4	Syn_0801_DefinitionOfAModule_003	A module definition with language and package is accepted.	Clause 8.1	m	
5	Syn_0801_DefinitionOfAModule_004	A module definition with package and without language is accepted.	Clause 8.1	m	
6	Syn_0801_DefinitionOfAModule_005	A module definition with ed4.3.1 language and package is accepted.	Clause 8.1	m	
7	Syn_0801_DefinitionOfAModule_006	A module definition with ed4.4.1 language and package is accepted.	Clause 8.1	m	

A.3.43 Module definitions part

Table A.42: 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	
2	Syn_0802_ModuleDefinitionsPart_002	A TypeDef module definition with private visibility is accepted.	Clause 8.2	m	

A.3.44 Module parameters

Table A.43: Module parameters

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	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	
2	Syn_080201_ModuleParameters_001	Plain module parameters are accepted.	Clause 8.2.1	m	
3	Syn_080201_ModuleParameters_002	Plain module parameters with default values are accepted.	Clause 8.2.1	m	
4	Syn_080201_ModuleParameters_003	Plain module parameters with default values and visibility modifiers are accepted.	Clause 8.2.1	m	

A.3.45 Groups of definitions

Table A.44: 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	
2	Syn_080202_GroupOfDefinitions_002	A definition within a nested group is accepted.	Clause 8.2.2	m	
3	Syn_080202_GroupOfDefinitions_003	A definition within a group with public visibility modifier is accepted.	Clause 8.2.2	m	
4	Syn_080202_GroupOfDefinitions_004	A definition within a group with public visibility modifier and attributes is accepted.	Clause 8.2.2	m	

A.3.46 General format of import

Table A.45: 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	
2	NegSem_08020301_GeneralFormatOfImport_002	Name handling of imported enumerations is properly handled.	Clause 8.2.3.1	m	
3	Syn_08020301_GeneralFormatOfImport_001	Import all is accepted.	Clause 8.2.3.1	m	
4	Syn_08020301_GeneralFormatOfImport_002	Import of specific types is accepted.	Clause 8.2.3.1	m	

A.3.47 Importing single definitions

Table A.46: 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	
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	

A.3.48 Importing groups

Table A.47: 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	
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	
3	Sem_08020303_ImportingGroups_002	The IUT properly handles 'except' clause in group import definitions.	Clause 8.2.3.3	m	
4	Sem_08020303_ImportingGroups_003	but that it is in fact a shortcut notation for explicit imports.	Clause 8.2.3.3	m	

A.3.49 Importing definitions of the same kind

Table A.48: 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	
2	NegSem_08020301_GeneralFormatOfImport_004	Transitive import rules are properly handled	Clause 8.2.3.4	m	
3	Sem_08020301_GeneralFormatOfImport_001	Transitive imports are properly handled	Clause 8.2.3.4	m	
4	Sem_08020301_GeneralFormatOfImport_002	Enumerated type definitions are automatically imported when needed	Clause 8.2.3.4	m	
5	Sem_08020304_ImportingDefinitionsOfTheSameKind_001	An import of all constants allows access to a sample constant.	Clause 8.2.3.4	m	
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	
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	

A.3.50 Importing all definitions of a module

Table A.49: 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	
2	Sem_08020305_ImportingAllDefinitionsOfAModule_001	The constant is be visible after multiple imports.	Clause 8.2.3.5	m	
3	Sem_08020305_ImportingAllDefinitionsOfAModule_002	The constant is be visible after multiple imports.	Clause 8.2.3.5	m	

A.3.51 Import definitions from other TTCN-3 editions and from non-TTCN-3 modules

Table A.50: 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	
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	
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	

A.3.52 Importing of import statements from TTCN-3 modules

Table A.51: 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	
2	NegSem_08020307_ImportingOfImportStatementsFromT3Modules_002	The import of import statements works for import all.	Clause 8.2.3.7	m	
3	Sem_08020307_ImportingOfImportStatementsFromT3Modules_001	The import of import statements works for import all.	Clause 8.2.3.7	m	

A.3.53 Compatibility of language specifications of imports

Table A.52: Compatibility of language specifications of imports

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_08020308_ImportingOfImportStatementsFromT3Modules_001	Imports referring to future TTCN-3 versions are rejected.	Clause 8.2.3.8	m	

A.3.54 Definition of friend modules

Table A.53: 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	
2	NegSem_080204_DefinitionOfFriendModules_002	Private definitions are not made visible by friend declarations (for a constant sample definition).	Clause 8.2.4	m	
3	Sem_080204_DefinitionOfFriendModules_001	Friend visibility works for a sample constant.	Clause 8.2.4	m	

A.3.55 Visibility of definitions

Table A.54: 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	
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	
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	
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	
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	
6	Sem_080205_VisibilityOfDefinitions_001	Explicitly defined public definitions (in this case a sample constant) are visible when imported.	Clause 8.2.5	m	
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	
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	
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	
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	

A.3.56 Module control part

Table A.55: 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	
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	
3	Syn_0803_ModuleControlPart_001	The module control is able to accept execute statements.	Clause 8.3	m	
4	Syn_0803_ModuleControlPart_002	The module control part with a few commonly used stateents is accepted.	Clause 8.3	m	
5	Syn_0803_ModuleControlPart_003	An empty control part is accepted.	Clause 8.3	m	

A.3.57 Port types, component types and test configurations

Table A.56: Port types, component types and test configurations

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_0901_Communication_ports_002	The IUT correctly handles loopback port connection to another port	Clause 9	m	
2	NegSem_0901_Communication_ports_003	The IUT correctly handles port connections	Clause 9	m	
3	Sem_0901_Communication_ports_001	The IUT correctly handles loopback message	Clause 9	m	
4	Sem_0901_Communication_ports_002	The the IUT receives the message sent by mycompA	Clause 9	m	
5	Sem_0901_Communication_ports_003	The the IUT receives the message sent by mycompB and mycompC	Clause 9	m	
6	Sem_0901_Communication_ports_004	The IUT correctly handles message exchange between ports	Clause 9	m	
7	Sem_0901_Communication_ports_005	The the IUT receives the message sent by mycompA	Clause 9	m	
8	NegSem_0902_Communication_ports_001	The IUT correctly handles the association of two ports to the same system interface	Clause 9	m	
9	NegSem_0902_Communication_ports_002	The mycomp is connected to two system interface port	Clause 9	m	
10	NegSem_0902_Communication_ports_003	The two system interface port cannot connect	Clause 9	m	
11	NegSem_0902_Communication_ports_004	The a connected port cannot be mapped	Clause 9	m	
12	Sem_0902_Communication_ports_001	The IUT port correctly mapped with a system interface	Clause 9	m	
13	Sem_0902_Communication_ports_002	The IUTs two ports are mapped correctly to system interfaces	Clause 9	m	
14	Syn_0902_Communication_ports_001	Two component can be mapped by one system interface	Clause 9	m	
15	NegSem_210102_disconnect_operation_001	Mapped port cannot disconnect	Clause 9	m	
16	Sem_210102_disconnect_operation_001	All component can be disconnected	Clause 9	m	
17	Sem_210102_disconnect_operation_002	Two components can disconnect	Clause 9	m	
18	Sem_210102_unmap_operation_001	unmap operation of two port	Clause 9	m	
19	Sem_210102_unmap_operation_002	unmap of system and MycompA	Clause 9	m	

A.3.58 Communication ports

Table A.57: Communication ports

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_0901_Communication_ports_001	The compiler handles the two port association with error	Clause 9.1	m	

A.3.59 Declaring constants

Table A.58: 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	
2	Sem_10_Constants_001	Assign and read constants	Clause 10	m	
3	Sem_10_Constants_002	Assign and read constants values	Clause 10	m	
4	Syn_10_Constants_001	Create constants	Clause 10	m	
5	Syn_10_Constants_002	Assign default constants values	Clause 10	m	
6	Syn_10_Constants_003	Assign component constants values	Clause 10	m	
7	Syn_10_Constants_004	Define constants in different scopes	Clause 10	m	

A.3.60 Value variables

Table A.59: 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	
2	NegSem_1101_ValueVars_002	Partially initialized variables are evaluated correctly.	Clause 11.1	m	
3	NegSyn_1101_ValueVars_001	Define variables in module scope.	Clause 11.1	m	
4	Sem_1101_ValueVars_001	Define variables in different scopes.	Clause 11.1	m	
5	Sem_1101_ValueVars_002	Define variables in different scopes.	Clause 11.1	m	
6	Sem_1101_ValueVars_003	Read and write variables.	Clause 11.1	m	
7	Sem_1101_ValueVars_004	Partially initialized variables are evaluated correctly.	Clause 11.1	m	
8	Sem_1101_ValueVars_005	Partially initialized variables are evaluated correctly.	Clause 11.1	m	
9	Syn_1101_ValueVars_001	Define variables in different scopes.	Clause 11.1	m	

A.3.61 Template variables

Table A.60: 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	
2	NegSem_1102_TemplateVars_002	Partially initialized templates are evaluated correctly.	Clause 11.2	m	
3	NegSyn_1102_TemplateVars_001	Define template variables in module scope.	Clause 11.2	m	
4	Sem_1102_TemplateVars_001	Define variables in different scopes	Clause 11.2	m	
5	Sem_1102_TemplateVars_002	Partially initialized templates are evaluated correctly.	Clause 11.2	m	
6	Sem_1102_TemplateVars_003	Partially initialized templates are evaluated correctly.	Clause 11.2	m	
7	Syn_1102_TemplateVars_001	Define template variables in different scopes.	Clause 11.2	m	

A.3.62 Declaring timers

Table A.61: 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	
2	NegSem_12_toplevel_timer_002	Ensure timer in array can not be initialized with negative duration	Clause 12	m	
3	NegSem_12_toplevel_timer_003	Ensure uninitialized timer can't be started	Clause 12	m	
4	NegSem_12_toplevel_timer_004	Ensure uninitialized timer in array can't be started	Clause 12	m	
5	NegSem_12_toplevel_timer_005	Ensure uninitialized timer in array can't be started	Clause 12	m	
6	NegSem_12_toplevel_timer_006	Ensure timer declaration syntax - reject single timer instance initialized with array	Clause 12	m	
7	NegSem_12_toplevel_timer_007	Ensure timer declaration syntax - reject array initialization with wrong number of initializers	Clause 12	m	
8	NegSem_12_toplevel_timer_008	Ensure timer declaration syntax - reject array of timers initizlized with a single float value	Clause 12	m	
9	NegSyn_12_toplevel_timer_001	Ensure timer can`t be used in module control parts when declared in components	Clause 12	m	
10	NegSyn_12_toplevel_timer_002	Ensure timer declaration syntax	Clause 12	m	
11	NegSyn_12_toplevel_timer_003	Ensure timer declaration syntax	Clause 12	m	
12	NegSyn_12_toplevel_timer_005	Ensure timer declaration syntax	Clause 12	m	
13	NegSyn_12_toplevel_timer_006	Ensure timer array declaration syntax	Clause 12	m	
14	NegSyn_12_toplevel_timer_007	Ensure timer array declaration syntax	Clause 12	m	
15	Sem_12_toplevel_timer_001	Ensure timer can be declared in components	Clause 12	m	
16	Sem_12_toplevel_timer_002	Ensure timer can be declared in module control parts	Clause 12	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
17	Sem_12_toplevel_timer_003	Ensure timer can be declared in altsteps	Clause 12	m	
18	Sem_12_toplevel_timer_004	Ensure timer can be declared in functions	Clause 12	m	
19	Sem_12_toplevel_timer_005	Ensure timer can be declared in test cases	Clause 12	m	
20	Sem_12_toplevel_timer_006	Ensure timer`s elapsed time is plausible	Clause 12	m	
21	Sem_12_toplevel_timer_007	Ensure timer can be declared in components but used in test cases	Clause 12	m	
22	Sem_12_toplevel_timer_008	Ensure timer can be declared in components but used in functions	Clause 12	m	
23	Sem_12_toplevel_timer_009	Ensure timer can be declared in components but used in altsteps	Clause 12	m	
24	Syn_12_toplevel_timer_001	Ensure non-initialized timer declaration syntax	Clause 12	m	
25	Syn_12_toplevel_timer_002	Ensure timer array declaration syntax	Clause 12	m	
26	Syn_12_toplevel_timer_003	Ensure definition of a list of timers is allowed as a single declaration	Clause 12	m	
27	Syn_12_toplevel_timer_004	Ensure timer array initialization syntax	Clause 12	m	
28	Syn_12_toplevel_timer_005	Ensure timer declaration with expression	Clause 12	m	
29	Syn_12_toplevel_timer_006	Ensure timer declaration with expression	Clause 12	m	
30	Sem_13_declaring_msg_001	Ensure received messages can be a combination of value and matching mechanism	Clause 12	m	

A.3.63 Declaring messages

Table A.62: Declaring messages

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	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	
2	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	
3	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	
4	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	
5	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	
6	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	
7	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	
8	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	
9	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	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
10	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	
11	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	
12	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	
13	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	
14	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	

A.3.64 Declaring procedure signatures

Table A.63: Declaring procedure signatures

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_1400_procedure_signatures_002	Blocking calls needs response or exception handling	Clause 14	m	
2	Sem_1400_procedure_signatures_001	The IUT calls signature exception	Clause 14	m	
3	Sem_1400_procedure_signatures_002	With noblock signature the IUT can raise exception	Clause 14	m	
4	Sem_1400_procedure_signatures_003	Non blocking signatures can raise exception	Clause 14	m	
5	Sem_1400_procedure_signatures_004	Multiple calls can be send without ack using non-blocking signature	Clause 14	m	

A.3.65 Declaring templates

Table A.64: 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	
2	Syn_15_TopLevel_001	A simple template with a single charstring field is accepted.	Clause 15	m	

A.3.66 Declaring message templates

Table A.65: 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	
2	Syn_1501_DeclaringMessageTemplates_002	A simple record-based message template with a wildcard ? is accepted	Clause 15.1	m	
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	
4	Syn_1501_DeclaringMessageTemplates_004	A primitive type template can be defined with a ? wildcard	Clause 15.1	m	
5	Syn_1501_DeclaringMessageTemplates_005	A primitive type template can be defined with a one-of notation	Clause 15.1	m	
6	Syn_1501_DeclaringMessageTemplates_006	All port operations are accepted	Clause 15.1	m	

A.3.67 Declaring signature templates

Table A.66: 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	
2	Sem_1502_DeclaringSignatureTemplates_002	Test in-line templates for accepting procedure replies.	Clause 15.2	m	
3	Sem_1502_DeclaringSignatureTemplates_003	Test in-line templates for accepting procedure replies.	Clause 15.2	m	
4	Syn_1502_DeclaringSignatureTemplates_001	Signature templates with explicit values are accepted.	Clause 15.2	m	
5	Syn_1502_DeclaringSignatureTemplates_002	Signature templates with wildcards are accepted.	Clause 15.2	m	
6	Syn_1502_DeclaringSignatureTemplates_003	The basic operations call and getreply are accepted.	Clause 15.2	m	
7	Syn_1502_DeclaringSignatureTemplates_004	The raise and catch operations are accepted.	Clause 15.2	m	

A.3.68 Global and local templates

Table A.67: Global and local templates

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_1503_GlobalAndLocalTemplates_001	A template values can be accessed with the dot notation as expected.	Clause 15.3	m	
2	Sem_1503_GlobalAndLocalTemplates_002	A template actual parameter is passed through correctly.	Clause 15.3	m	
3	Sem_1503_GlobalAndLocalTemplates_003	A send operation with actual parameters of a global parameterized template is accepted.	Clause 15.3	m	
4	Sem_1503_GlobalAndLocalTemplates_004	A parameterized local template in a test case is accepted.	Clause 15.3	m	
5	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	
6	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	
7	Syn_1503_GlobalAndLocalTemplates_001	A global parameterized template is accepted.	Clause 15.3	m	
8	Syn_1503_GlobalAndLocalTemplates_004	A parameterized local template in the control part is accepted.	Clause 15.3	m	
9	Syn_1503_GlobalAndLocalTemplates_005	A parameterized local template in a function is accepted.	Clause 15.3	m	
10	Syn_1503_GlobalAndLocalTemplates_006	A parameterized local template in an altstep is accepted.	Clause 15.3	m	

A.3.69 In-line templates

Table A.68: 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	
2	Syn_1504_InlineTemplates_002	Modified parameterized inline templates are accepted.	Clause 15.4	m	
3	Syn_1504_InlineTemplates_003	Modified plain inline templates are accepted.	Clause 15.4	m	

A.3.70 Modified templates

Table A.69: 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	
2	NegSem_1505_ModifiedTemplates_002	A modified template does not omit possible parameters of the base template.	Clause 15.5	m	
3	NegSem_1505_ModifiedTemplates_003	A modified template does not omit possible parameters introduced in any modification step.	Clause 15.5	m	
4	NegSem_1505_ModifiedTemplates_004	Parameter names in modified templates are the same.	Clause 15.5	m	
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	
6	NegSem_1505_ModifiedTemplates_006	The same parameter name is used when modifying the base template.	Clause 15.5	m	
7	NegSem_1505_ModifiedTemplates_007	The same parameter type is used when modifying the base template.	Clause 15.5	m	
8	Sem_1505_ModifiedTemplates_001	The values of plain modified template definitions are as expected.	Clause 15.5	m	
9	Sem_1505_ModifiedTemplates_002	A modified template of a record of type using index notation access works as expected.	Clause 15.5	m	
10	Sem_1505_ModifiedTemplates_003	Default values in formal parameters of modified templates are working as expected.	Clause 15.5	m	
11	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	
12	Syn_1505_ModifiedTemplates_001	Plain modified template definitions are accepted.	Clause 15.5	m	
13	Syn_1505_ModifiedTemplates_002	A modified template does not omit possible parameters introduced in any modification step.	Clause 15.5	m	
14	Syn_1505_ModifiedTemplates_003	The default values in formal parameters of modified templates are accepted.	Clause 15.5	m	
15	Syn_1505_ModifiedTemplates_004	Dash as default parameter values are accepted.	Clause 15.5	m	

A.3.71 Referencing individual string elements

Table A.70: 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	

A.3.72 Referencing record and set fields

Table A.71: 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	
2	NegSem_150602_ReferencingRecordAndSetFields_002	Fields with * values on the right-hand side of an assignment are rejected	Clause 15.6.2	m	
3	NegSem_150602_ReferencingRecordAndSetFields_003	Value lists on the right-hand side of an assignment are not acceped.	Clause 15.6.2	m	
4	NegSem_150602_ReferencingRecordAndSetFields_004	Complement lists on the right-hand side of an assignment are not acceped	Clause 15.6.2	m	
5	NegSem_150602_ReferencingRecordAndSetFields_005	Referencing a template field with the ifpresent attribute causes a rejection	Clause 15.6.2	m	
6	Sem_150602_ReferencingRecordAndSetFields_001	? shall be returned for mandatory subfields and * shall be returned for optional subfields	Clause 15.6.2	m	
7	Sem_150602_ReferencingRecordAndSetFields_002	The recurisve anyvalue expansion is performed correctly when new values are assigned.	Clause 15.6.2	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
8	Sem_150602_ReferencingRecordAndSetFields_003	? shall be returned for mandatory subfields and * shall be returned for optional subfields	Clause 15.6.2	m	
9	Sem_150602_ReferencingRecordAndSetFields_004	? shall be returned for mandatory subfields and * shall be returned for optional subfields	Clause 15.6.2	m	

A.3.73 Referencing record of and set of elements

Table A.72: 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	
2	NegSem_150603_ReferencingRecordOfAndSetElements_002	Access to unitialized fields in the context of record of is rejected.	Clause 15.6.3	m	
3	NegSem_150603_ReferencingRecordOfAndSetElements_003	Anyvalueornone fields in the context of record of is rejected.	Clause 15.6.3	m	
4	NegSem_150603_ReferencingRecordOfAndSetElements_004	Complement value lists in the context of record of are rejected.	Clause 15.6.3	m	
5	NegSem_150603_ReferencingRecordOfAndSetElements_005	Subset in the context of record of are rejected.	Clause 15.6.3	m	
6	NegSem_150603_ReferencingRecordOfAndSetElements_006	Superset in the context of record of are rejected.	Clause 15.6.3	m	
7	NegSem_150603_ReferencingRecordOfAndSetElements_007	Access into permutation in record of templates is forbidden.	Clause 15.6.3	m	
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	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
9	NegSem_150603_ReferencingRecordOfAndSetElements_009	Access to ifpresent fields is not allowed.	Clause 15.6.3	m	
10	NegSem_150603_ReferencingRecordOfAndSetElements_010	Referencing AnyValueOrNone fields is not allowed.	Clause 15.6.3	m	
11	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	
12	Sem_150603_ReferencingRecordOfAndSetElements_002	Assignment to a anyvalue in the context of record of is handled correctly.	Clause 15.6.3	m	
13	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	
14	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	
15	Sem_150603_ReferencingRecordOfAndSetElements_005	Access outside permutation fields is allowed and works as expected.	Clause 15.6.3	m	

A.3.74 Template restrictions

Table A.73: 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	
2	NegSem_1508_TemplateRestrictions_002	Template(omit) is rejected with setof template.	Clause 15.8	m	
3	NegSem_1508_TemplateRestrictions_003	Template(omit) is rejected with anyvalueornone(*).	Clause 15.8	m	
4	NegSem_1508_TemplateRestrictions_004	Template(omit) is rejected with value ranges.	Clause 15.8	m	
5	NegSem_1508_TemplateRestrictions_005	Template(omit) is rejected with supersets.	Clause 15.8	m	
6	NegSem_1508_TemplateRestrictions_006	Template(omit) is rejected with subsets.	Clause 15.8	m	
7	NegSem_1508_TemplateRestrictions_007	Template(omit) is rejected with patterns.	Clause 15.8	m	
8	NegSem_1508_TemplateRestrictions_008	Template(omit) is rejected with anyelement inside values.	Clause 15.8	m	
9	NegSem_1508_TemplateRestrictions_009	Template(omit) is rejected with anyelemenornone inside values.	Clause 15.8	m	
10	NegSem_1508_TemplateRestrictions_010	Template(omit) is rejected with permutation inside values.	Clause 15.8	m	
11	NegSem_1508_TemplateRestrictions_011	Template(omit) is rejected with length restrictions.	Clause 15.8	m	
12	NegSem_1508_TemplateRestrictions_012	Template(omit) is rejected with length restrictions.	Clause 15.8	m	
13	NegSem_1508_TemplateRestrictions_013	Template(omit) is rejected with length restrictions.	Clause 15.8	m	
14	NegSem_1508_TemplateRestrictions_014	Template(value) is rejected with anyvalue(?).	Clause 15.8	m	
15	NegSem_1508_TemplateRestrictions_015	Template(value) is rejected with valuelist.	Clause 15.8	m	
16	NegSem_1508_TemplateRestrictions_016	Template(value) is rejected with anyvalueornone(*).	Clause 15.8	m	
17	NegSem_1508_TemplateRestrictions_017	Template(value) is rejected with value ranges.	Clause 15.8	m	
18	NegSem_1508_TemplateRestrictions_018	Template(value) is rejected with supersets.	Clause 15.8	m	
19	NegSem_1508_TemplateRestrictions_019	Template(value) is rejected with supersets.	Clause 15.8	m	
20	NegSem_1508_TemplateRestrictions_020	Template(value) is rejected with patterns.	Clause 15.8	m	
21	NegSem_1508_TemplateRestrictions_021	Template(value) is rejected with anyelement inside values.	Clause 15.8	m	
22	NegSem_1508_TemplateRestrictions_022	Template(value) is rejected with permutation inside values.	Clause 15.8	m	
23	NegSem_1508_TemplateRestrictions_023	Template(value) is rejected with length restrictions.	Clause 15.8	m	
24	NegSem_1508_TemplateRestrictions_024	Template(value) is rejected with length restrictions.	Clause 15.8	m	
25	NegSem_1508_TemplateRestrictions_025	Template(present) refuses omitvalue as a whole.	Clause 15.8	m	
26	NegSem_1508_TemplateRestrictions_026	Template(value) refuses omit as a whole.	Clause 15.8	m	
27	Sem_1508_TemplateRestrictions_001	A value can be assigned to a template(omit) variable.	Clause 15.8	m	
28	Sem_1508_TemplateRestrictions_002	A template(omit) can be assigned to a template(omit) variable.	Clause 15.8	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
29	Sem_1508_TemplateRestrictions_003	A templat(value) can be assigned to a template(omit) variable.	Clause 15.8	m	
30	Sem_1508_TemplateRestrictions_004	A value can be assigned to a template(value) variable.	Clause 15.8	m	
31	Sem_1508_TemplateRestrictions_005	A template(value) can be assigned to a template(value) variable.	Clause 15.8	m	
32	Sem_1508_TemplateRestrictions_006	A value can be assigned to a template(present) variable.	Clause 15.8	m	
33	Sem_1508_TemplateRestrictions_007	A template(omit) can be assigned to a template(present) variable.	Clause 15.8	m	
34	Sem_1508_TemplateRestrictions_008	A template(value) can be assigned to a template(present) variable.	Clause 15.8	m	
35	Sem_1508_TemplateRestrictions_009	A template(present) can be assigned to a template(present) variable.	Clause 15.8	m	
36	Sem_1508_TemplateRestrictions_010	A value can be assigned to a template variable.	Clause 15.8	m	
37	Sem_1508_TemplateRestrictions_011	A template(omit) can be assigned to a template variable.	Clause 15.8	m	
38	Sem_1508_TemplateRestrictions_012	A template(value) can be assigned to a template variable.	Clause 15.8	m	
39	Sem_1508_TemplateRestrictions_013	A template(present) can be assigned to a template variable.	Clause 15.8	m	
40	Sem_1508_TemplateRestrictions_014	A template can be assigned to a template variable.	Clause 15.8	m	
41	Sem_1508_TemplateRestrictions_015	A base template can be modified without restrictions.	Clause 15.8	m	
42	Syn_1508_TemplateRestrictions_001	Template(omit) is accepted with value omitvalue.	Clause 15.8	m	
43	Syn_1508_TemplateRestrictions_002	Template(omit) is accepted with a concrete value.	Clause 15.8	m	
44	Syn_1508_TemplateRestrictions_003	Template(value) is accepted with a concrete value.	Clause 15.8	m	
45	Syn_1508_TemplateRestrictions_004	Template(present) is accepted with a concrete value.	Clause 15.8	m	

A.3.75 Match operation

Table A.74: 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	
2	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	
3	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	
4	Sem_1509_MatchOperation_003	The match operation works correctly on records in the positive case.	Clause 15.9	m	
5	Sem_1509_MatchOperation_004	The match operation works correctly on records in the negative case.	Clause 15.9	m	
6	Sem_1509_MatchOperation_005	The match operation works correctly if the types are incompatible.	Clause 15.9	m	

A.3.76 Valueof operation

Table A.75: 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	
2	NegSem_1510_ValueOfOperation_002	The valueof function works correctly on templates with wildcards.	Clause 15.10	m	
3	NegSem_1510_ValueOfOperation_003	The valueof function works correctly on regular value templates.	Clause 15.10	m	
4	NegSem_1510_ValueOfOperation_004	The valueof function works correctly on range templates.	Clause 15.10	m	
5	Sem_1510_ValueOfOperation_001	The valueof operation works as expected for fully initialized templates.	Clause 15.10	m	

A.3.77 Concatenating templates of string and list types

Table A.76: 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	
2	NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_002	Concatenation of strings types yields an error if specified ranges are not fixed length.	Clause 15.11	m	
3	NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_003	A simple concatenation of non-wildcard octetstring shall not yield in a non-even number of hexadecimals.	Clause 15.11	m	
4	NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_004	The inline template definitions are correctly concatenated.	Clause 15.11	m	
5	NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_005	The inline template definitions are correctly concatenated.	Clause 15.11	m	
6	NegSem_1511_ConcatenatingTemplatesOfStringAndListTypes_006	Concatenation of octetstring types and ? patterns works as expected.	Clause 15.11	m	
7	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_001	Concatenation of charstring types works as expected (variant 1).	Clause 15.11	m	
8	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_002	Concatenation of octetstring types works as expected (variant 2).	Clause 15.11	m	
9	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_003	Concatenation of bitstring types works as expected.	Clause 15.11	m	
10	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_004	Concatenation of octetstring types works as expected (variant 1).	Clause 15.11	m	
11	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_005	Concatenation of octetstring types works as expected (variant 2).	Clause 15.11	m	
12	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_006	A concatenation of charstrings with a fixed length AnyValueOrNone be matched.	Clause 15.11	m	
13	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_008	Concatenations of record of charstrings work when parameterized.	Clause 15.11	m	
14	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_009	Concatenations of set of integers are accepted.	Clause 15.11	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
15	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_010	The inline template definitions are correctly concatenated.	Clause 15.11	m	
16	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_011	Concatenation of octetstring types works as expected (matching patterns in quotation).	Clause 15.11	m	
17	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_012	Concatenation of octetstring types and ? patterns works as expected.	Clause 15.11	m	
18	Sem_1511_ConcatenatingTemplatesOfStringAndListTypes_013	Concatenation of octetstring types and ? patterns works as expected.	Clause 15.11	m	

A.3.78 Functions

Table A.77: 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	
2	NegSem_1601_toplevel_002	The IUT correctly handles function definitions	Clause 16.1	m	
3	NegSem_1601_toplevel_003	The IUT correctly handles function definitions	Clause 16.1	m	
4	NegSem_1601_toplevel_004	The IUT correctly handles function definitions	Clause 16.1	m	
5	NegSem_1601_toplevel_005	The IUT correctly handles function definitions	Clause 16.1	m	
6	NegSem_1601_toplevel_006	The IUT correctly handles function definitions	Clause 16.1	m	
7	Sem_1601_toplevel_001	The IUT correctly handles function definitions	Clause 16.1	m	
8	Sem_1601_toplevel_003	The IUT correctly handles function definitions	Clause 16.1	m	

A.3.79 Invoking functions

Table A.78: 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	

A.3.80 Predefined functions

Table A.79: 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	
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	
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	
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	
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	
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	
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	
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	
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	
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	
11	NegSem_160102_predefined_functions_011	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
12	NegSem_160102_predefined_functions_014	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
13	NegSem_160102_predefined_functions_015	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
14	NegSem_160102_predefined_functions_016	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
15	NegSem_160102_predefined_functions_017	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
16	NegSem_160102_predefined_functions_018	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
17	NegSem_160102_predefined_functions_019	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
18	NegSem_160102_predefined_functions_020	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C.33)	Clause 16.1.2	m	
19	NegSem_160102_predefined_functions_021	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
20	NegSem_160102_predefined_functions_022	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
21	NegSem_160102_predefined_functions_023	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
22	NegSem_160102_predefined_functions_024	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
23	NegSem_160102_predefined_functions_025	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
24	NegSem_160102_predefined_functions_026	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
25	NegSem_160102_predefined_functions_027	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
26	NegSem_160102_predefined_functions_028	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
27	NegSem_160102_predefined_functions_029	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
28	NegSem_160102_predefined_functions_030	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
29	NegSem_160102_predefined_functions_031	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
30	Sem_160102_predefined_functions_001	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
31	Sem_160102_predefined_functions_002	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
32	Sem_160102_predefined_functions_003	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
33	Sem_160102_predefined_functions_004	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
34	Sem_160102_predefined_functions_005	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
35	Sem_160102_predefined_functions_006	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
36	Sem_160102_predefined_functions_007	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
37	Sem_160102_predefined_functions_008	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
38	Sem_160102_predefined_functions_009	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
39	Sem_160102_predefined_functions_010	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
40	Sem_160102_predefined_functions_011	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
41	Sem_160102_predefined_functions_012	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
42	Sem_160102_predefined_functions_013	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
43	Sem_160102_predefined_functions_014	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	
44	Sem_160102_predefined_functions_015	The IUT recognizes predefined functions and correctly evaluates them (as specified by Annex C)	Clause 16.1.2	m	

A.3.81 External functions

Table A.80: 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	
2	NegSem_160103_external_functions_002	Port parameters cannot be passed to external functions as inout parameters	Clause 16.1.3	m	
3	NegSem_160103_external_functions_003	Timer parameters cannot be passed to external functions as inout parameters	Clause 16.1.3	m	
4	Sem_160103_external_functions_001	The IUT recognizes external functions	Clause 16.1.3	m	
5	Sem_160103_external_functions_002	The IUT recognizes external functions	Clause 16.1.3	m	

A.3.82 Invoking function from specific places

Table A.81: 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	
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	
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	
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	

A.3.83 Altsteps

Table A.82: 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	
2	NegSem_1602_toplevel_002	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
3	NegSem_1602_toplevel_003	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
4	NegSem_1602_toplevel_004	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
5	NegSem_1602_toplevel_005	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
6	NegSem_1602_toplevel_006	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
7	NegSyn_1602_toplevel_001	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	
8	Sem_1602_toplevel_001	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2	m	

A.3.84 Invoking altsteps

Table A.83: 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	
2	Sem_160201_invoking_altsteps_001	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2.1	m	
3	Sem_160201_invoking_altsteps_002	The IUT recognizes altstep definitions and correctly evaluates them	Clause 16.2.1	m	
4	Sem_160201_invoking_altsteps_003	Altsteps are correctly handled for dynamically mapped ports	Clause 16.2.1	m	
5	Sem_160201_invoking_altsteps_004	Altsteps are correctly handled for dynamically mapped ports	Clause 16.2.1	m	

A.3.85 Test cases

Table A.84: 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	
2	NegSem_1603_testcases_002	The IUT properly evaluates invocation of testcases	Clause 16.3	m	
3	Syn_1603_testcases_001	The IUT properly evaluates invocation of testcases with system clause	Clause 16.3	m	

A.3.86 Assignments

Table A.85: 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	
2	NegSem_1901_assignments_002	The IUT properly evaluates assignment statements	Clause 19.1	m	
3	NegSem_1901_assignments_003	The IUT properly evaluates assignment statements	Clause 19.1	m	
4	NegSyn_1901_assignments_001	The IUT properly evaluates assignment statements	Clause 19.1	m	
5	Sem_1901_assignments_001	The IUT properly evaluates assignment statements	Clause 19.1	m	

A.3.87 The if-else statement

Table A.86: 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	
2	Sem_1902_if_else_statement_001	The IUT properly evaluates if-else statements	Clause 19.2	m	
3	Sem_1902_if_else_statement_002	The IUT properly evaluates if-else statements	Clause 19.2	m	

A.3.88 The select case statement

Table A.87: The select case statement

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_1903_select_case_statement_001	The IUT properly evaluates select-case statements	Clause 19.3	m	
2	Sem_1903_select_case_statement_002	The IUT properly evaluates select-case statements	Clause 19.3	m	
3	Sem_1903_select_case_statement_003	The IUT properly evaluates select-case statements	Clause 19.3	m	
4	Sem_1903_select_case_statement_004	The IUT properly evaluates select-case statements	Clause 19.3	m	

A.3.89 The for statement

Table A.88: 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	
2	Sem_1904_for_statement_001	The IUT properly evaluates for statements	Clause 19.4	m	
3	Sem_1904_for_statement_002	The IUT properly evaluates for statements	Clause 19.4	m	
4	Sem_1904_for_statement_003	The IUT properly evaluates for statements	Clause 19.4	m	

A.3.90 The while statement

Table A.89: 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	
2	Sem_1905_while_statement_001	The IUT properly evaluates while statements	Clause 19.5	m	
3	Sem_1905_while_statement_002	The IUT properly evaluates while statements	Clause 19.5	m	
4	Sem_1905_while_statement_003	The IUT properly evaluates while statements	Clause 19.5	m	

A.3.91 The do-while statement

Table A.90: 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	
2	Sem_1906_do_while_statement_001	The IUT properly evaluates do-while statements	Clause 19.6	m	
3	Sem_1906_do_while_statement_002	The IUT properly evaluates do-while statements	Clause 19.6	m	
4	Sem_1906_do_while_statement_003	The IUT properly evaluates do-while statements	Clause 19.6	m	

A.3.92 The label statement

Table A.91: 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	
2	NegSyn_1907_label_statement_001	The IUT correctly handles label syntax.	Clause 19.7	m	
3	NegSyn_1907_label_statement_002	The IUT correctly handles label syntax.	Clause 19.7	m	
4	Syn_1907_label_statement_001	The IUT correctly handles label syntax.	Clause 19.7	m	

A.3.93 The goto statement

Table A.92: 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	
2	NegSem_1908_goto_statement_002	The IUT correctly handles goto statements.	Clause 19.8	m	
3	NegSem_1908_goto_statement_003	The IUT correctly handles goto statements.	Clause 19.8	m	
4	Sem_1908_goto_statement_001	The IUT correctly handles goto statements.	Clause 19.8	m	
5	Sem_1908_goto_statement_002	The IUT correctly handles goto statements.	Clause 19.8	m	
6	Sem_1908_goto_statement_003	The IUT correctly handles goto statements.	Clause 19.8	m	

A.3.94 The stop execution statement

Table A.93: 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	
2	Sem_1909_stop_statement_002	The IUT correctly handles stop statements.	Clause 19.9	m	

A.3.95 The return statement

Table A.94: 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	
2	Sem_1910_return_statement_001	The IUT correctly handles return statements.	Clause 19.10	m	
3	Sem_1910_return_statement_002	The IUT correctly handles return statements.	Clause 19.10	m	

A.3.96 The log statement

Table A.95: 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	
2	Sem_1911_log_statement_001	The IUT properly evaluates log statements	Clause 19.11	m	
3	Sem_1911_log_statement_002	The IUT properly evaluates log statements	Clause 19.11	m	
4	Sem_1911_log_statement_003	The IUT properly evaluates log statements	Clause 19.11	m	
5	Sem_1911_log_statement_004	The IUT properly evaluates log statements	Clause 19.11	m	
6	Sem_1911_log_statement_005	The IUT properly evaluates log statements	Clause 19.11	m	

A.3.97 The continue statement

Table A.96: 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	

A.3.98 Statement and operations for alternative behaviours

Table A.97: 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	
2	Syn_20_TopLevel_002	Repeat in an alt-statement is accepted.	Clause 20	m	
3	Syn_20_TopLevel_003	The interleave-statement is accepted.	Clause 20	m	
4	Syn_20_TopLevel_004	Defaults and the activate statement is accepted.	Clause 20	m	
5	Syn_20_TopLevel_005	Defaults and the activate statement is accepted.	Clause 20	m	

A.3.99 The alt statement

Table A.98: The alt statement

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_2002_TheAltStatement_001	The alt-statement works as expected (loopback case).	Clause 20.2	m	
2	Sem_2002_TheAltStatement_002	The alt-statement with a guard works as expected (loopback case).	Clause 20.2	m	
3	Sem_2002_TheAltStatement_003	The alt-statement processes the alternatives in order (loopback case).	Clause 20.2	m	
4	Sem_2002_TheAltStatement_004	Activated defaults are processed in the reverse order (loopback case).	Clause 20.2	m	
5	Sem_2002_TheAltStatement_005	The else branch is executed when nothing else matched (loopback case).	Clause 20.2	m	
6	Sem_2002_TheAltStatement_006	An altstep invocation works as expected (loopback case).	Clause 20.2	m	
7	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	
8	Sem_2002_TheAltStatement_008	The done-block in an alt-statement is triggered as expected (loopback case).	Clause 20.2	m	
9	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	
10	Sem_2002_TheAltStatement_010	The timeout branch is taken as expected (loopback case).	Clause 20.2	m	
11	Sem_2002_TheAltStatement_011	The behavior continues after the alt-statement (loopback case).	Clause 20.2	m	
12	Sem_2002_TheAltStatement_012	Alt statements are correctly handled for dynamically mapped ports	Clause 20.2	m	
13	Sem_2002_TheAltStatement_013	Alt statements are correctly handled for dynamically mapped ports	Clause 20.2	m	

A.3.100 The repeat statement

Table A.99: 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	
2	Sem_2003_the_repeat_statement_001	The IUT correctly processes repeat statements	Clause 20.3	m	

A.3.101 The interleave statement

Table A.100: 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	
2	NegSyn_2004_InterleaveStatement_001	Validate that interleave statements are properly handled.	Clause 20.4	m	
3	NegSyn_2004_InterleaveStatement_002	Validate that interleave statements are properly handled.	Clause 20.4	m	
4	Sem_2004_InterleaveStatement_001	Validate that interleave statements are properly handled.	Clause 20.4	m	
5	Sem_2004_InterleaveStatement_002	Validate that interleave statements are properly handled.	Clause 20.4	m	
6	Syn_2004_InterleaveStatement_001	Validate that interleave statements are properly handled.	Clause 20.4	m	

A.3.102 Configuration operations

Table A.101: Configuration operations

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_210101_connect_operation_001	The the IUT does not allows two output port connection	Clause 21	m	
2	NegSem_210101_connect_operation_002	The the IUT does not allow connecting incompatible ports	Clause 21	m	

A.3.103 Connection operations

Table A.102: Connection operations

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_210101_map_operation_001	IUT cannot map input port with output port	Clause 21.1	m	
2	NegSem_210101_map_operation_002	IUT cannot map input port with output port	Clause 21.1	m	

A.3.104 Test case operations

Table A.103: 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	

A.3.105 The create operation

Table A.104: 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	
2	NegSem_210301_CreateOperation_002	Named components on hosts are accepted	Clause 21.3.1	m	
3	NegSem_210301_CreateOperation_003	Named components on hosts are accepted	Clause 21.3.1	m	
4	Sem_210301_CreateOperation_001	Unnamed components can be created	Clause 21.3.1	m	
5	Sem_210301_CreateOperation_002	Named components can be created	Clause 21.3.1	m	
6	Sem_210301_CreateOperation_003	Unnamed alive components on hosts can be created	Clause 21.3.1	m	
7	Sem_210301_CreateOperation_004	Named alive components can be created	Clause 21.3.1	m	
8	Syn_210301_CreateOperation_001	Named components on hosts are accepted	Clause 21.3.1	m	

A.3.106 The start test component operation

Table A.105: 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	
2	NegSem_210302_Start_test_component_002	Only component type is allowed for ptc declaration	Clause 21.3.2	m	
3	Sem_210302_Start_test_component_001	Alive test components are allowed to start another function	Clause 21.3.2	m	

A.3.107 The stop test behaviour operation

Table A.106: The stop test behaviour operation

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_210303_Stop_test_component_001	Component.stop causes the stopping of the target component	Clause 21.3.3	m	
2	Sem_210303_Stop_test_component_002	Self.stop stops current component	Clause 21.3.3	m	

A.3.108 The kill test component operation

Table A.107: The kill test component operation

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_210304_kill_test_component_001	Kill operator stops a non alive test component	Clause 21.3.4	m	
2	Sem_210304_kill_test_component_002	All component kill stop all ptcs	Clause 21.3.4	m	
3	Sem_210304_kill_test_component_003	Kill operator stops only non alive test components	Clause 21.3.4	m	
4	Sem_210304_kill_test_component_004	Self kill called in a functions stops non alive test component	Clause 21.3.4	m	

A.3.109 The alive operation

Table A.108: The alive operation

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_210305_alive_operation_001	Testing alive operator with an alive test component	Clause 21.3.5	m	
2	Sem_210305_alive_operation_002	Test all component alive operator with alive test components	Clause 21.3.5	m	
3	Sem_210305_alive_operation_003	Alive operator gives a correct boolean result	Clause 21.3.5	m	
4	Sem_210305_alive_operation_004	Test any component alive operator with multiple test components	Clause 21.3.5	m	

A.3.110 The running operation

Table A.109: The running operation

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_210306_running_operation_001	Check that running operator provides information about test components.	Clause 21.3.6	m	
2	Sem_210306_running_operation_002	Any component with running can check the status of the test components	Clause 21.3.6	m	

A.3.111 The done operation

Table A.110: 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	
2	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	

A.3.112 The killed operation

Table A.111: 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	
2	Sem_210308_killed_operation_001	All component kill can be checked with killed operator	Clause 21.3.8	m	
3	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	
4	Sem_210308_killed_operation_003	Ensure that the alive keyword is properly evaluated	Clause 21.3.8	m	

A.3.113 The send operation

Table A.112: 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	
2	NegSem_220201_SendOperation_002	The IUT correctly handles message sending operations	Clause 22.2.1	m	
3	NegSem_220201_SendOperation_003	The IUT correctly handles message sending operations	Clause 22.2.1	m	
4	NegSem_220201_SendOperation_004	The IUT correctly handles message sending operations	Clause 22.2.1	m	
5	Sem_220201_SendOperation_001	The IUT correctly handles message sending operations	Clause 22.2.1	m	
6	Sem_220201_SendOperation_002	The IUT correctly handles message sending operations	Clause 22.2.1	m	
7	Sem_220201_SendOperation_003	The IUT correctly handles message sending operations	Clause 22.2.1	m	
8	Sem_220201_SendOperation_004	The IUT correctly handles message sending operations	Clause 22.2.1	m	

A.3.114 The receive operation

Table A.113: 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	
2	Sem_220202_ReceiveOperation_001	The IUT correctly handles message receiving operations	Clause 22.2.2	m	
3	Sem_220202_ReceiveOperation_002	The IUT correctly handles message receiving operations	Clause 22.2.2	m	
4	Sem_220202_ReceiveOperation_003	The IUT correctly handles message receiving operations	Clause 22.2.2	m	
5	Sem_220202_ReceiveOperation_004	The IUT correctly handles message receiving operations	Clause 22.2.2	m	
6	Sem_220202_ReceiveOperation_005	The IUT correctly handles message receiving operations	Clause 22.2.2	m	

A.3.115 The trigger operation

Table A.114: 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	
2	Sem_220203_TriggerOperation_001	The IUT correctly handles message trigger operations	Clause 22.2.3	m	
3	Sem_220203_TriggerOperation_002	The IUT correctly handles message trigger operations	Clause 22.2.3	m	
4	Sem_220203_TriggerOperation_003	The IUT correctly handles message trigger operations	Clause 22.2.3	m	
5	Sem_220203_TriggerOperation_004	The IUT correctly handles message trigger operations	Clause 22.2.3	m	
6	Sem_220203_TriggerOperation_005	The IUT correctly handles message trigger operations	Clause 22.2.3	m	

A.3.116 The call operation

Table A.115: The call operation

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 22.3.1	m	
2	NegSem_220301_CallOperation_001	The IUT correctly handles procedure call operations	Clause 22.3.1	m	
3	NegSem_220301_CallOperation_002	The IUT correctly procedure calls	Clause 22.3.1	m	
4	Sem_220301_CallOperation_001	The IUT correctly handles procedure call operations	Clause 22.3.1	m	
5	Sem_220301_CallOperation_002	The IUT correctly handles procedure call operations	Clause 22.3.1	m	
6	Sem_220301_CallOperation_003	The IUT correctly handles non-blocking procedure call	Clause 22.3.1	m	
7	Sem_220301_CallOperation_004	The IUT correctly handles multiple client calls to the same server	Clause 22.3.1	m	
8	Sem_220301_CallOperation_005	The IUT correctly handles broadcast/multicast procedure call	Clause 22.3.1	m	
9	Sem_220301_CallOperation_006	The IUT correctly handles broadcast/multicast procedure call	Clause 22.3.1	m	
10	Sem_220301_CallOperation_007	The IUT correctly handles blocking procedure call	Clause 22.3.1	m	

A.3.117 The getcall operation

Table A.116: 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	
2	NegSem_220302_GetcallOperation_002	Getcall operation does not allow value assignment	Clause 22.3.2	m	
3	NegSem_220302_GetcallOperation_003	Getcall for any call does not allow param assignment	Clause 22.3.2	m	
4	Sem_220302_GetcallOperation_001	Getcall operations remove only matching procedure from the queue	Clause 22.3.2	m	
5	Sem_220302_GetcallOperation_002	Getcall operations remove the matching procedure from the queue	Clause 22.3.2	m	
6	Sem_220302_GetcallOperation_003	The getcall operation can be correctly restricted to a certain client	Clause 22.3.2	m	
7	Sem_220302_GetcallOperation_004	The getcall operation can be correctly restricted to a certain client	Clause 22.3.2	m	
8	Sem_220302_GetcallOperation_005	Getcall operations work with any port attribute	Clause 22.3.2	m	

A.3.118 The reply operation

Table A.117: 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	
2	Sem_220303_ReplyOperation_001	The IUT correctly handles reply to multiple clients on the same server	Clause 22.3.3	m	
3	Sem_220303_ReplyOperation_002	The IUT correctly handles reply to multiple clients on the same server	Clause 22.3.3	m	

A.3.119 Timer operations

Table A.118: Timer operations

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	m	
2	NegSem_2302_timer_start_002	Ensure not_a_number is not allowed	Clause 23	m	
3	NegSem_2302_timer_start_003	Ensure negative value is not allowed	Clause 23	m	
4	NegSem_2302_timer_start_004	Ensure negative infinity is not allowed	Clause 23	m	
5	NegSyn_2302_timer_start_001	Ensure timer start syntax	Clause 23	m	
6	NegSyn_2302_timer_start_002	Ensure timer start syntax	Clause 23	m	
7	NegSyn_2302_timer_start_003	Ensure timer start syntax	Clause 23	m	
8	NegSyn_2302_timer_start_004	Ensure timer start syntax	Clause 23	m	
9	NegSyn_2302_timer_start_005	Ensure timer start syntax	Clause 23	m	
10	NegSyn_2302_timer_start_006	Ensure timer start syntax	Clause 23	m	
11	NegSyn_2302_timer_start_007	Ensure timer start syntax	Clause 23	m	
12	NegSyn_2302_timer_start_008	Ensure timer start syntax	Clause 23	m	
13	NegSyn_2302_timer_start_009	Ensure timer start syntax	Clause 23	m	
14	NegSyn_2302_timer_start_010	Ensure timer start syntax	Clause 23	m	
15	NegSyn_2302_timer_start_011	Ensure timer start syntax	Clause 23	m	
16	NegSyn_2302_timer_start_012	Ensure timer start syntax	Clause 23	m	
17	NegSyn_2302_timer_start_013	Ensure timer start syntax	Clause 23	m	
18	Sem_2302_timer_start_001	Ensure timer runs from zero to stated value	Clause 23	m	
19	Sem_2302_timer_start_002	Ensure timer can be restarted	Clause 23	m	
20	Sem_2302_timer_start_003	Ensure timer default value can be modified by start value	Clause 23	m	
21	Sem_2302_timer_start_004	Ensure timer with value 0.0 expires immediately	Clause 23	m	
22	NegSem_23_toplevel_001	Ensure timer operations are not allowed outside of module control, test case, function, altstep	Clause 23	m	
23	NegSem_23_toplevel_002	Ensure timer operations are not allowed outside of module control, test case, function, altstep	Clause 23	m	
24	NegSyn_23_toplevel_001	Ensure timer operations are not allowed outside of module control, test case, function, altstep	Clause 23	m	
25	NegSyn_23_toplevel_002	Ensure timer operations are not allowed outside of module control, test case, function, altstep	Clause 23	m	
26	Syn_23_toplevel_001	Ensure timer allowed in module control, test case, function, altstep	Clause 23	m	
27	Syn_23_toplevel_002	Ensure timer allowed in module control, test case, function, altstep	Clause 23	m	

A.3.120 The stop timer operation

Table A.119: 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	
2	NegSyn_2303_timer_stop_002	Ensure timer stop syntax	Clause 23.3	m	
3	NegSyn_2303_timer_stop_003	Ensure all timer stop syntax	Clause 23.3	m	
4	NegSyn_2303_timer_stop_004	Ensure all timer stop syntax	Clause 23.3	m	
5	NegSyn_2303_timer_stop_005	Ensure all timer stop syntax	Clause 23.3	m	
6	NegSyn_2303_timer_stop_006	Ensure all timer stop syntax	Clause 23.3	m	
7	Sem_2303_timer_stop_002	Ensure timer stop sets elapsed time to zero	Clause 23.3	m	
8	Sem_2303_timer_stop_003	Ensure timer all timer identifier	Clause 23.3	m	
9	Sem_2303_timer_stop_004	Ensure can be stopped after timeout	Clause 23.3	m	
10	Syn_2303_timer_stop_006	Ensure timer stop syntax	Clause 23.3	m	
11	Syn_2303_timer_stop_007	Ensure all timer stop syntax	Clause 23.3	m	

A.3.121 The running timer operation

Table A.120: 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	
2	NegSyn_2305_timer_running_002	Ensure timer running syntax	Clause 23.5	m	
3	NegSyn_2305_timer_running_003	Ensure timer running syntax	Clause 23.5	m	
4	NegSyn_2305_timer_running_004	Ensure timer running syntax	Clause 23.5	m	
5	NegSyn_2305_timer_running_005	Ensure timer running syntax	Clause 23.5	m	
6	NegSyn_2305_timer_running_006	Ensure timer running syntax: disallow all timer.running	Clause 23.5	m	
7	Sem_2305_timer_running_001	Ensure timer running any timer identifier works	Clause 23.5	m	
8	Sem_2305_timer_running_002	Ensure timer running operation works	Clause 23.5	m	
9	Sem_2305_timer_running_003	Ensure timer running operation works	Clause 23.5	m	
10	Sem_2305_timer_running_004	Ensure timer running operation works	Clause 23.5	m	
11	Syn_2306_timer_timeout_001	Ensure timer runnig syntax	Clause 23.5	m	

A.3.122 The timeout operation

Table A.121: 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	
2	NegSyn_2306_timer_timeout_002	Ensure timer timeout cannot be used in boolean expressions	Clause 23.6	m	
3	NegSyn_2306_timer_timeout_003	Ensure timer timeout syntax	Clause 23.6	m	
4	NegSyn_2306_timer_timeout_004	Ensure timer timeout syntax	Clause 23.6	m	
5	NegSyn_2306_timer_timeout_005	Ensure timer timeout syntax	Clause 23.6	m	
6	NegSyn_2306_timer_timeout_006	Ensure timer timeout syntax	Clause 23.6	m	
7	NegSyn_2306_timer_timeout_007	Ensure timer timeout syntax	Clause 23.6	m	
8	Sem_2306_timer_timeout_001	Ensure timer timeout operations: non-started timer does not timeout	Clause 23.6	m	
9	Sem_2306_timer_timeout_002	Ensure timer timeout operations: timed-out timer does not timeout until restarted	Clause 23.6	m	
10	Sem_2306_timer_timeout_003	Ensure timer timeout happen in order from the shortest to the longest	Clause 23.6	m	
11	Sem_2306_timer_timeout_004	Ensure any timer.timeout operation	Clause 23.6	m	
12	Sem_2306_timer_timeout_005	Ensure any timer.timeout operation for timeouts that are not in scope	Clause 23.6	m	
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	
14	Sem_2306_timer_timeout_007	Ensure timer timeout happen in order from the shortest to the longest	Clause 23.6	m	

A.3.123 Test verdict operations

Table A.122: Test verdict operations

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_13_declaring_msg_002	Ensure received messages cannot be matched with wrong template	Clause 24	m	
2	Sem_13_declaring_msg_003	Ensure instances of messages can be declared by in-line templates	Clause 24	m	
3	Sem_13_declaring_msg_004	Ensure instances of messages can be declared by global templates	Clause 24	m	
4	Sem_13_declaring_msg_005	Ensure instances of messages can be declared and passed via template variables	Clause 24	m	
5	Sem_13_declaring_msg_006	Ensure instances of messages can be declared and passed via inline template	Clause 24	m	
6	Sem_13_declaring_msg_007	Ensure instances of messages can be declared and passed via parameter	Clause 24	m	
7	Sem_13_declaring_msg_008	Ensure instances of messages can be declared and passed via template parameter	Clause 24	m	
8	Sem_13_declaring_msg_009	Ensure instances of messages can be declared and passed via template parameter	Clause 24	m	
9	NegSem_2402_setverdict_params_001	Ensure setverdict accepts parameters of verdicttype only	Clause 24	m	
10	NegSem_2402_setverdict_params_002	Ensure setverdict accepts parameters of verdicttype only	Clause 24	m	
11	NegSem_2402_setverdict_params_003	Ensure setverdict accepts values of verdicttype only	Clause 24	m	
12	NegSem_2402_setverdict_params_004	Ensure setverdict accepts values only as the parameter	Clause 24	m	
13	NegSem_2402_setverdict_params_005	Ensure setverdict accepts values only as the parameter	Clause 24	m	
14	Sem_2402_setverdict_logging_001	Ensure logging constraints	Clause 24	m	
15	Sem_2402_setverdict_params_001	Ensure setverdict accepts values only as the parameter	Clause 24	m	
16	Sem_2402_setverdict_params_002	Ensure setverdict accepts values only as the parameter	Clause 24	m	
17	Sem_2402_setverdict_params_003	Ensure logging contraints	Clause 24	m	
18	NegSem_24_toplevel_001	Ensure getverdict is not allowed in constant initialization in control part	Clause 24	m	
19	NegSem_24_toplevel_002	Ensure getverdict is not allowed in parameter initialization in control part	Clause 24	m	
20	NegSem_24_toplevel_003	Ensure getverdict is not allowed in variable definition in control part.	Clause 24	m	
21	NegSem_24_toplevel_004	Ensure setverdict is not allowed in part whithin compound statement	Clause 24	m	
22	NegSem_24_toplevel_005	Ensure setverdict is not allowed in control part at the top level	Clause 24	m	
23	Syn_24_toplevel_001	Ensure setverdict and getverdict are allowed in functions	Clause 24	m	
24	Syn_24_toplevel_002	Ensure setverdict and getverdict are allowed in test cases	Clause 24	m	
25	Syn_24_toplevel_003	Ensure setverdict and getverdict are allowed in atsteps	Clause 24	m	

A.3.124 The verdict mechanism

Table A.123: The verdict mechanism

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_2401_SetverdictError	Setverdict cannot set error verdict	Clause 24.1	m	
2	Sem_2401_GlobalVerdict_001	Ensure overwriting rules for global verdict: pass can overwrite none.	Clause 24.1	m	
3	Sem_2401_GlobalVerdict_002	Ensure overwriting rules for global verdict: inconc can overwrite none.	Clause 24.1	m	
4	Sem_2401_GlobalVerdict_003	Ensure overwriting rules for global verdict: fail can overwrite none.	Clause 24.1	m	
5	Sem_2401_GlobalVerdict_004	Ensure overwriting rules for global verdict: none cannot overwrite pass.	Clause 24.1	m	
6	Sem_2401_GlobalVerdict_005	Ensure overwriting rules for global verdict: inconc can overwrite pass.	Clause 24.1	m	
7	Sem_2401_GlobalVerdict_006	Ensure overwriting rules for global verdict: fail can overwrite pass.	Clause 24.1	m	
8	Sem_2401_GlobalVerdict_007	Ensure overwriting rules for global verdict: none cannot overwrite inconc.	Clause 24.1	m	
9	Sem_2401_GlobalVerdict_008	Ensure overwriting rules for global verdict: pass cannot overwrite inconc.	Clause 24.1	m	
10	Sem_2401_GlobalVerdict_009	Ensure overwriting rules for global verdict: fail can overwrite inconc.	Clause 24.1	m	
11	Sem_2401_GlobalVerdict_010	Ensure overwriting rules for global verdict: none cannot overwrite fail.	Clause 24.1	m	
12	Sem_2401_GlobalVerdict_011	Ensure overwriting rules for global verdict: pass cannot overwrite fail.	Clause 24.1	m	
13	Sem_2401_GlobalVerdict_012	Ensure overwriting rules for global verdict: inconc cannot overwrite fail.	Clause 24.1	m	
14	Sem_2401_InitiallyNone_001	Local verdicts initializes with none	Clause 24.1	m	
15	Sem_2401_LocalVerdict_001	Ensure overwriting rules for local verdict: pass can overwrite none.	Clause 24.1	m	
16	Sem_2401_LocalVerdict_002	Ensure overwriting rules for local verdict: inconc can overwrite none.	Clause 24.1	m	
17	Sem_2401_LocalVerdict_003	Ensure overwriting rules for local verdict: fail can overwrite none.	Clause 24.1	m	
18	Sem_2401_LocalVerdict_004	Ensure overwriting rules for local verdict: none cannot overwrite pass.	Clause 24.1	m	
19	Sem_2401_LocalVerdict_005	Ensure overwriting rules for local verdict: inconc can overwrite pass.	Clause 24.1	m	
20	Sem_2401_LocalVerdict_006	Ensure overwriting rules for local verdict: fail can overwrite pass.	Clause 24.1	m	
21	Sem_2401_LocalVerdict_007	Ensure overwriting rules for local verdict: none cannot overwrite inconc.	Clause 24.1	m	
22	Sem_2401_LocalVerdict_008	Ensure overwriting rules for local verdict: pass cannot overwrite inconc.	Clause 24.1	m	
23	Sem_2401_LocalVerdict_009	Ensure overwriting rules for local verdict: fail can overwrite inconc.	Clause 24.1	m	
24	Sem_2401_LocalVerdict_010	Ensure overwriting rules for local verdict: none cannot overwrite fail.	Clause 24.1	m	
25	Sem_2401_LocalVerdict_011	Ensure overwriting rules for local verdict: pass cannot overwrite fail.	Clause 24.1	m	
26	Sem_2401_LocalVerdict_012	Ensure overwriting rules for local verdict: inconc cannot overwrite fail.	Clause 24.1	m	
27	Syn_2401_FiveValues_001	There are five values of verdicttype	Clause 24.1	m	

A.3.125 The getverdict mechanism

Table A.124: 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	
2	Sem_2403_getverdict_002	Ensure getverdict returns the actual verdict inconc	Clause 24.3	m	
3	Sem_2403_getverdict_003	Ensure getverdict returns the actual verdict pass	Clause 24.3	m	
4	Sem_2403_getverdict_004	Ensure getverdict returns the actual verdict fail	Clause 24.3	m	
5	Sem_2403_getverdict_005	Ensure getverdict none for uninitialized verdict	Clause 24.3	m	

A.3.126 Module control

Table A.125: 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	
2	Syn_26_ModuleControl_002	If-else constructs in the control part are accepted.	Clause 26	m	
3	Syn_26_ModuleControl_003	Select-case constructs in the control part are accepted.	Clause 26	m	
4	Syn_26_ModuleControl_004	For loop constructs in the control part are accepted.	Clause 26	m	
5	Syn_26_ModuleControl_005	While loop constructs in the control part are accepted.	Clause 26	m	
6	Syn_26_ModuleControl_006	Label and goto constructs in the control part are accepted.	Clause 26	m	
7	Syn_26_ModuleControl_007	The stop construct in the control part is accepted.	Clause 26	m	
8	Syn_26_ModuleControl_008	The break construct in the control part is accepted.	Clause 26	m	
9	Syn_26_ModuleControl_009	The continue construct in the control part is accepted.	Clause 26	m	
10	Syn_26_ModuleControl_010	The continue construct in the control part is accepted.	Clause 26	m	
11	Syn_26_ModuleControl_011	The alt/timeout construct in the control part is accepted.	Clause 26	m	
12	Syn_26_ModuleControl_012	The repeat construct in the control part is accepted.	Clause 26	m	
13	Syn_26_ModuleControl_013	The interleave construct in the control part is accepted.	Clause 26	m	
14	Syn_26_ModuleControl_014	Activate/deactivate/default constructs in the control part are accepted.	Clause 26	m	
15	Syn_26_ModuleControl_015	Start/stop/read/running timer constructs in the control part are accepted.	Clause 26	m	
16	Syn_26_ModuleControl_016	The action construct in the control part is accepted.	Clause 26	m	
17	Syn_26_ModuleControl_017	The execute construct in the control part is accepted.	Clause 26	m	

A.3.127 The execute statement

Table A.126: 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	
2	NegSem_2601_ExecuteStatement_002	Non-float timeout parameters in the execute statement are rejected (in this case charstring).	Clause 26.1	m	
3	NegSem_2601_ExecuteStatement_003	Host id can be only charstring.	Clause 26.1	m	
4	NegSem_2601_ExecuteStatement_004	Execution rejects test case execution with infinity timer guard.	Clause 26.1	m	
5	Sem_2601_ExecuteStatement_001	Parameters are passed correctly into the test case.	Clause 26.1	m	
6	Sem_2601_ExecuteStatement_002	Multiple parameters of different types are passed correctly into the test case.	Clause 26.1	m	
7	Sem_2601_ExecuteStatement_003	The timeout specified with the execute statement is respected.	Clause 26.1	m	
8	Sem_2601_ExecuteStatement_004	The verdict none works correctly.	Clause 26.1	m	
9	Sem_2601_ExecuteStatement_005	The verdict pass works correctly.	Clause 26.1	m	
10	Sem_2601_ExecuteStatement_006	The verdict inconc works correctly.	Clause 26.1	m	
11	Sem_2601_ExecuteStatement_007	The timeout specified with the execute statement is respected.	Clause 26.1	m	
12	Sem_2601_ExecuteStatement_008	The user error sets the verdict error correctly.	Clause 26.1	m	
13	Sem_2601_ExecuteStatement_009	Host id restriction is correctly handled.	Clause 26.1	m	

A.3.128 The control part

Table A.127: 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	
2	NegSem_2602_TheControlPart_002	The create component is not allowed in the control part.	Clause 26.2	m	
3	NegSem_2602_TheControlPart_003	The create alive component is not allowed in the control part.	Clause 26.2	m	
4	NegSem_2602_TheControlPart_004	The start statement is not allowed in the control part.	Clause 26.2	m	
5	NegSem_2602_TheControlPart_005	The stop statement is not allowed in the control part.	Clause 26.2	m	
6	NegSem_2602_TheControlPart_006	The kill statement is not allowed in the control part.	Clause 26.2	m	
7	NegSem_2602_TheControlPart_007	The alive operation is not allowed in the control part.	Clause 26.2	m	
8	NegSem_2602_TheControlPart_008	The running operation is not allowed in the control part.	Clause 26.2	m	
9	NegSem_2602_TheControlPart_009	The done operation is not allowed in the control part.	Clause 26.2	m	
10	NegSem_2602_TheControlPart_010	The killed operation is not allowed in the control part.	Clause 26.2	m	
11	NegSem_2602_TheControlPart_011	The connect statements are not allowed in the control part.	Clause 26.2	m	
12	NegSem_2602_TheControlPart_012	The disconnect statements are not allowed in the control part.	Clause 26.2	m	
13	NegSem_2602_TheControlPart_013	The map statements are not allowed in the control part.	Clause 26.2	m	
14	NegSem_2602_TheControlPart_014	The unmap statements are not allowed in the control part.	Clause 26.2	m	
15	NegSem_2602_TheControlPart_015	The send statements are not allowed in the control part.	Clause 26.2	m	
16	NegSem_2602_TheControlPart_016	The receive statements are not allowed in the control part.	Clause 26.2	m	
17	NegSem_2602_TheControlPart_017	The call statements are not allowed in the control part.	Clause 26.2	m	
18	NegSem_2602_TheControlPart_018	The reply statements are not allowed in the control part.	Clause 26.2	m	
19	NegSem_2602_TheControlPart_019	The raise statements are not allowed in the control part.	Clause 26.2	m	
20	NegSem_2602_TheControlPart_020	The trigger statements are not allowed in the control part.	Clause 26.2	m	
21	NegSem_2602_TheControlPart_021	The getcall statements are not allowed in the control part.	Clause 26.2	m	
22	NegSem_2602_TheControlPart_022	The getreply statements are not allowed in the control part.	Clause 26.2	m	
23	NegSem_2602_TheControlPart_023	The catch statements are not allowed in the control part.	Clause 26.2	m	
24	NegSem_2602_TheControlPart_024	The check statements are not allowed in the control part.	Clause 26.2	m	
25	NegSem_2602_TheControlPart_025	The clear statements are not allowed in the control part.	Clause 26.2	m	
26	NegSem_2602_TheControlPart_026	The start statements on ports are not allowed in the control part.	Clause 26.2	m	
27	NegSem_2602_TheControlPart_027	The stop statements on ports are not allowed in the control part.	Clause 26.2	m	
28	NegSem_2602_TheControlPart_028	The halt statements are not allowed in the control part.	Clause 26.2	m	
29	NegSem_2602_TheControlPart_029	Alternative behaviours are only used to control timer behavior in the control part.	Clause 26.2	m	

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
30	NegSem_2602_TheControlPart_030	Getverdict statements are not allowed in the control part.	Clause 26.2	m	
31	NegSem_2602_TheControlPart_031	Execute statements are not executed from test cases.	Clause 26.2	m	
32	NegSem_2602_TheControlPart_032	The create alive named component is not allowed in the control part.	Clause 26.2	m	
33	NegSem_2602_TheControlPart_033	The create named component is not allowed in the control part.	Clause 26.2	m	
34	NegSem_2602_TheControlPart_034	The create named component on host is not allowed in the control part.	Clause 26.2	m	
35	NegSem_2602_TheControlPart_035	Alternative behaviours are only used to control timer behavior in the control part.	Clause 26.2	m	
36	Sem_2602_TheControlPart_001	The selection/deselection of test cases using boolean conditions works as expected.	Clause 26.2	m	
37	Sem_2602_TheControlPart_002	The execution of test cases works from within a function.	Clause 26.2	m	
38	Sem_2602_TheControlPart_003	The selection of test cases can be achieven based on resulting verdict types.	Clause 26.2	m	

A.3.129 Scope of attributes

Table A.128: 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	
2	Syn_270101_ScopeOfAttributes_002	Attributes for language elements are accepted.	Clause 27.1.1	m	
3	Syn_270101_ScopeOfAttributes_003	Attributes for individual fields are accepted.	Clause 27.1.1	m	
4	Syn_270101_ScopeOfAttributes_004	Attributes for individual fields are accepted.	Clause 27.1.1	m	

A.3.130 Optional attributes

Table A.129: Optional attributes

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	NegSem_2707_OptionalAttributes_002	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
2	NegSem_2707_OptionalAttributes_003	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
3	Sem_2707_OptionalAttributes_001	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
4	Sem_2707_OptionalAttributes_002	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
5	Sem_2707_OptionalAttributes_003	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
6	Sem_2707_OptionalAttributes_004	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
7	Sem_2707_OptionalAttributes_005	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
8	Sem_2707_OptionalAttributes_006	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
9	Sem_2707_OptionalAttributes_007	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
10	Sem_2707_OptionalAttributes_008	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	
11	Syn_2707_OptionalAttributes_001	The IUT correctly handles attribute definitions and their scoping rules	Clause 27.7	m	

A.3.131 Matching specific values

Table A.130: 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	
2	Sem_B0101_matching_specific_value_002	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
3	Sem_B0101_matching_specific_value_003	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
4	Sem_B0101_matching_specific_value_004	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
5	Sem_B0101_matching_specific_value_005	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
6	Sem_B0101_matching_specific_value_006	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
7	Sem_B0101_matching_specific_value_007	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
8	Sem_B0101_matching_specific_value_008	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
9	Sem_B0101_matching_specific_value_009	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
10	Sem_B0101_matching_specific_value_010	The IUT correctly handles template matching of specific values	Clause B.1.1	m	
11	Sem_B0101_matching_specific_value_011	The IUT correctly handles template matching of specific values	Clause B.1.1	m	

A.3.132 Value list

Table A.131: Value list

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_B010201_value_list_001	The IUT correctly handles template matching of listed multiple values	Clause B.1.2.1	m	

A.3.133 Complemented value list

Table A.132: Complemented value list

Item	TC/TP reference	Purpose	Reference in ES 201 873-1	Status	Support
1	Sem_B010202_complemented_value_list_001	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	
2	Sem_B010202_complemented_value_list_002	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	
3	Sem_B010202_complemented_value_list_003	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	
4	Sem_B010202_complemented_value_list_004	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	
5	Sem_B010202_complemented_value_list_005	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	
6	Sem_B010202_complemented_value_list_006	The IUT correctly handles template matching of complemented value listing	Clause B.1.2.2	m	

A.3.134 Any value

Table A.133: 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	
2	Sem_B010203_any_value_002	The IUT correctly handles template matching of ? values	Clause B.1.2.3	m	

A.3.135 Any value or none

Table A.134: 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	
2	NegSem_B010204_any_value_or_none_002	The IUT correctly handles template matching of * values	Clause B.1.2.4	m	
3	Sem_B010204_any_value_or_none_001	The IUT correctly handles template matching of * values	Clause B.1.2.4	m	

A.3.136 Value range

Table A.135: 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	
2	NegSem_B010205_value_range_002	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
3	NegSem_B010205_value_range_003	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
4	Sem_B010205_value_range_001	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
5	Sem_B010205_value_range_002	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
6	Sem_B010205_value_range_003	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
7	Sem_B010205_value_range_004	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
8	Sem_B010205_value_range_005	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
9	Sem_B010205_value_range_006	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
10	Sem_B010205_value_range_007	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	
11	Sem_B010205_value_range_008	The IUT correctly handles template matching of value range definitions	Clause B.1.2.5	m	

A.3.137 SuperSet

Table A.136: 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	
2	NegSem_B010206_superset_002	The IUT correctly handles template matching of superset definitions	Clause B.1.2.6	m	
3	Sem_B010206_superset_001	The IUT correctly handles template matching of superset definitions	Clause B.1.2.6	m	
4	Sem_B010206_superset_002	The IUT correctly handles template matching of superset definitions	Clause B.1.2.6	m	
5	Sem_B010206_superset_003	The IUT correctly handles template matching of superset definitions	Clause B.1.2.6	m	

A.3.138 SubSet

Table A.137: 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 superset definitions	Clause B.1.2.7	m	
2	NegSem_B010207_subset_002	The IUT correctly handles template matching of superset definitions	Clause B.1.2.7	m	
3	Sem_B010207_subset_001	The IUT correctly handles template matching of superset definitions	Clause B.1.2.7	m	
4	Sem_B010207_subset_002	The IUT correctly handles template matching of superset definitions	Clause B.1.2.7	m	
5	Sem_B010207_subset_003	The IUT correctly handles template matching of superset definitions	Clause B.1.2.7	m	

A.3.139 Any element

Table A.138: 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	
2	Sem_B010301_any_element_002	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
3	Sem_B010301_any_element_003	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
4	Sem_B010301_any_element_004	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
5	Sem_B010301_any_element_005	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
6	Sem_B010301_any_element_006	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
7	Sem_B010301_any_element_007	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	
8	Sem_B010301_any_element_008	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.1	m	

A.3.140 Any number of elements of no element

Table A.139: 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	
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	
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	

A.3.141 Permutation

Table A.140: 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	
2	Sem_B010303_permutation_001	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.3	m	
3	Sem_B010303_permutation_002	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.3	m	
4	Sem_B010303_permutation_003	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.3	m	
5	Sem_B010303_permutation_004	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.3	m	
6	Sem_B010303_permutation_005	The IUT correctly handles template matching of ? symbols in value elements	Clause B.1.3.3	m	
7	Sem_B010303_permutation_006	The IUT correctly handles permutation within arrays	Clause B.1.3.3	m	

A.3.142 Length restrictions

Table A.141: 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	
2	NegSem_B010401_length_restrictions_002	The IUT correctly handles template matching of value length definitions	Clause B.1.4.1	m	
3	NegSem_B010401_length_restrictions_003	The IUT correctly handles template matching of value length definitions	Clause B.1.4.1	m	
4	NegSem_B010401_length_restrictions_004	The IUT correctly handles template matching of value length definitions	Clause B.1.4.1	m	
5	Sem_B010401_length_restrictions_001	The IUT correctly handles template matching of value length definitions	Clause B.1.4.1	m	
6	Sem_B010401_length_restrictions_002	The IUT correctly handles template matching of value length definitions	Clause B.1.4.1	m	

A.3.143 The ifpresent indicator

Table A.142: 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	
2	Sem_B010402_ifPresent_indicator_001	The IUT correctly handles template matching of ifpresent indicators	Clause B.1.4.2	m	
3	Sem_B010402_ifPresent_indicator_002	The IUT correctly handles template matching of ifpresent indicators	Clause B.1.4.2	m	

A.3.144 Matching character pattern

Table A.143: 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	

A.3.145 Set expression

Table A.144: 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	
2	Sem_B010501_set_expression_001	The IUT correctly handles template matching of character pattern set expressions	Clause B.1.5.1	m	
3	Sem_B010501_set_expression_002	The IUT correctly handles template matching of character pattern set expressions	Clause B.1.5.1	m	
4	Sem_B010501_set_expression_003	The IUT correctly handles template matching of character pattern set expressions	Clause B.1.5.1	m	
5	Sem_B010501_set_expression_004	The IUT correctly handles template matching of character pattern set expressions	Clause B.1.5.1	m	
6	Sem_B010501_set_expression_005	The IUT correctly handles template matching of character pattern set expressions	Clause B.1.5.1	m	

A.3.146 Reference expression

Table A.145: 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	
2	Sem_B010502_reference_expression_002	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
3	Sem_B010502_reference_expression_003	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
4	Sem_B010502_reference_expression_004	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
5	Sem_B010502_reference_expression_005	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
6	Sem_B010502_reference_expression_006	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
7	Sem_B010502_reference_expression_007	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
8	Sem_B010502_reference_expression_008	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	
9	Sem_B010502_reference_expression_009	The IUT correctly handles template matching of character pattern reference expressions	Clause B.1.5.2	m	

A.3.147 Match expression n times

Table A.146: 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	
2	Sem_B010503_match_n_times_002	The IUT correctly handles template matching of character pattern expression multiplicity	Clause B.1.5.3	m	
3	Sem_B010503_match_n_times_003	The IUT correctly handles template matching of character pattern expression multiplicity	Clause B.1.5.3	m	
4	Sem_B010503_match_n_times_004	The IUT correctly handles template matching of character pattern expression multiplicity	Clause B.1.5.3	m	
5	Sem_B010503_match_n_times_005	The IUT correctly handles template matching of character pattern expression multiplicity	Clause B.1.5.3	m	

A.3.148 Match a referenced character set

Table A.147: 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	
2	Sem_B010504_match_referenced_characters_001	The IUT correctly handles template matching of character pattern reference characters	Clause B.1.5.4	m	
3	Sem_B010504_match_referenced_characters_002	The IUT correctly handles template matching of character pattern reference characters	Clause B.1.5.4	m	
4	Sem_B010504_match_referenced_characters_003	The IUT correctly handles template matching of character pattern reference characters	Clause B.1.5.4	m	
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	
7	Sem_B010504_match_referenced_characters_006	The IUT correctly handles template matching of character pattern reference characters	Clause B.1.5.4	m	
8	Sem_B010504_match_referenced_characters_007	The IUT correctly handles template matching of character pattern reference characters	Clause B.1.5.4	m	

A.3.149 Type compatibility rules for patterns

Table A.148: Type compatibility rules for patterns

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

A.3.150 Preprocessing macros

Table A.149: 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	
2	Sem_D02_macro_file_001	__FILE__ macro stores the path and file name in a charstring	Clause D	m	
3	Sem_D03_macro_bfile_001	The __BFILE__ macro replaces the actual file name	Clause D	m	
4	Sem_D04_macro_line_001	__LINE__ macro stores the actual line number when it is called	Clause D	m	
5	NegSem_D05_macro_scope_001	__SCOPE__ replaces the actual higher named basic scope unit	Clause D	m	
6	Sem_D05_macro_scope_001	__SCOPE__ replaces the actual higher basic unit	Clause D	m	
7	Sem_D05_macro_scope_002	__SCOPE__ replaces the actual higher basic unit	Clause D	m	

A.4 Additional information for ICS

This clause contains all additional comments provided by the supplier of the implementation.

History

Document history		
V1.1.1	April 2011	Publication
V1.2.1	April 2012	Publication
V1.3.1	October 2013	Publication