

# Statement of Compliance for use of XML Schema in Eclipse Titan

Adrien Kirjak

Version 11.2.1, 2026-08-05

# Table of Contents


1. Description . . . 2
2. References . . . 3
   2.1. Normative references . . . 3
   2.2. Informative references . . . 3
3. Definitions and abbreviations . . . 4
   3.1. Definitions . . . 4
   3.2. Abbreviations . . . 4
4. Instructions for completing the ICS proforma . . . 6
   4.1. Other information . . . 6
   4.2. Identification of the implementation . . . 7
5. ICS proforma tables . . . 9
   5.1. Global statement of conformance . . . 9
   5.2. Mapping XML Schemas . . . 9
   5.3. Namespaces . . . 9
   5.4. Includes . . . 10
   5.5. Imports . . . 10
   5.6. Attributes of the XSD schema element . . . 10
   5.7. Name conversion rules . . . 12
   5.8. Order of the mapping . . . 15
   5.9. Built-in data types . . . 15
   5.10. Length . . . 16
   5.11. Enumeration . . . 16
   5.12. MinInclusive . . . 17
   5.13. MaxInclusive . . . 18
   5.14. MinExclusive . . . 18
   5.15. MaxExclusive . . . 19
   5.16. Total digits . . . 20
   5.17. Fraction digits . . . 21
   5.18. Not specifically mapped facets . . . 21
   5.19. String . . . 21
   5.20. Name . . . 21
   5.21. Any URI . . . 22
   5.22. Integer . . . 22
   5.23. Positive integer . . . 22
   5.24. Non-positive integer . . . 23
   5.25. Negative integer . . . 23
   5.26. Non-negative integer . . . 23
   5.27. Long . . . 23

5.28. Unsigned long . . . 24
5.29. Int . . . 24
5.30. Unsigned int . . . 24
5.31. Short . . . 25
5.32. Unsigned Short . . . 25
5.33. Byte . . . 25
5.34. Unsigned byte . . . 25
5.35. Decimal . . . 26
5.36. Float . . . 26
5.37. Double . . . 26
5.38. Date and time . . . 27
5.39. Date . . . 27
5.40. Gregorian year and month . . . 28
5.41. Gregorian year . . . 29
5.42. Boolean type . . . 30
5.43. AnyType and anySimpleType types . . . 31
5.44. Id . . . 31
5.45. MinOccurs and maxOccurs . . . 32
5.46. Default and Fixed . . . 33
5.47. Form . . . 34
5.48. Type . . . 36
5.49. Use . . . 36
5.50. Final . . . 37
5.51. Element component . . . 37
5.52. Attribute element definitions . . . 37
5.53. Attribute group definitions . . . 38
5.54. Derivation by restriction . . . 38
5.55. Derivation by list . . . 38
5.56. Derivation by union . . . 39
5.57. Extending simple content . . . 40
5.58. Restricting simple content . . . 41
5.59. Complex content derived by extension . . . 41
5.60. Complex content derived by restriction . . . 43
5.61. Referencing group components . . . 43
5.62. All content . . . 45
5.63. Choice content . . . 45
5.64. Choice with nested elements . . . 46
5.65. Choice with nested group . . . 46
5.66. Choice with nested choice . . . 46
5.67. Choice with nested sequence . . . 47
5.68. Choice with nested any . . . 47

5.69. Sequence with nested element content . . . 48
5.70. Sequence with nested group content . . . 48
5.71. Sequence with nested choice content . . . 48
5.72. Sequence with nested sequence content . . . 48
5.73. Sequence with nested any content . . . 49
5.74. Effect of the minOccurs and maxOccurs attributes on the mapping . . . 49
5.75. Attribute definitions, attribute and attributeGroup references . . . 50
5.76. Mixed content . . . 51
5.77. The any element . . . 52
5.78. The anyAttribute element . . . 53
5.79. Annotation . . . 54
5.80. Group components . . . 54
5.81. Identity-constraint definition schema components . . . 55
5.82. Head elements of substitution groups . . . 55
5.83. TTCN-3 module XSD . . . 56
6. Notes . . . 67

## Abstract

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

### Copyright

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

### Disclaimer

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

# Chapter 1. Description

The present document provides the Implementation Conformance Statement (ICS) proforma for the conformance test suite for using XML Schema with TTCN-3 as defined in ETSI ES 201 873-1.

In the present document only XML related features, specified in ETSI ES 201 873 9 have been considered but not

- the core language features (see ETSI ES 201 873-1), nor
- tool implementation (see ETSI ES 201 873-5 and ETSI ES 201 873-6),
- language mapping (see ETSI ES 201 873-7 and ETSI ES 201 873-8) and language extension (see e.g. ETSI ES 202 781, ETSI ES 202 784 and ETSI ES 202 785) aspects.

# Chapter 2. References

## 2.1. Normative references

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

1. ETSI ES 201 873-9 (V4.6.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3".
2. ISO/IEC 9646-7 (1994): "Conformance testing methodology and framework – Part 7: Implementation Conformance Statement".
3. ISO/IEC 9646-1 (1992): "Information Technology – Open Systems Interconnection – Conformance Testing Methodology and Framework – Part 1: General concepts".
4. ETSI ES 202 785: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types".

## 2.2. Informative references

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

5. ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".
6. 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)".
7. 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").
8. 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".
9. 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".
10. ETSI ES 202 781: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Configuration and Deployment Support".
11. ETSI ES 202 784: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization".

# Chapter 3. Definitions and abbreviations

## 3.1. Definitions

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

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

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

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

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

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

## 3.2. Abbreviations

**ASCI**

American Standard Code for Information Interchange

**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**

Testing and Test Control Notation

**TTCN-3**

Testing and Test Control Notation edition 3

**URI**

Uniform Resource Identifier

**URL**

Uniform Resource Locator

**XML**

eXtensible Markup Language

**XSD**

W3C XML Schema Definition

# Chapter 4. Instructions for completing the ICS proforma

## 4.1. Other information

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

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

### 4.1.1. Purposes and structure

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

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

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

### 4.1.2. Conventions

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

**Item column**

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

**Item description column**

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

**Reference column**

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

**Status column**

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

- m mandatory - the capability is required to be supported.
- n/a not applicable - in the given context, it is impossible to use the capability. No answer in the support column is required.
- u undecided
- o optional - the capability may be supported or not.
- o.i qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer

which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table.

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

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

**Support column**

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

- Y or y supported by the implementation.
- N or n not supported by the implementation.
- N/A or n/a or "no answer required" (allowed only if the status is N/A, directly or after evaluation of a conditional status).

**Values allowed column**

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

**Values supported column**

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

**References to items**

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

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

## 4.2. Identification of the implementation

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

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

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

### 4.2.1. Date of the statement

| **Date of the statement:** | **2016.07.19** |
|---|---|

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

| **IUT name:** | **Eclipse Titan** |
|---|---|
| IUT version: | CRL 113 200/5 R5B |

### 4.2.3. ICS contact person

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

# Chapter 5. ICS proforma tables

## 5.1. Global statement of conformance

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

## 5.2. Mapping XML Schemas

*Table 1. Mapping XML Schemas*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_05_top_level_001 | Verify that error is generated for missing XSD language tag in import clause | Clause 5 | m | n |

## 5.3. Namespaces

*Table 2. Namespaces*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_050101_namespaces_001 | Verify that schema with target namespace is correctly translated into single module | Clause 5.1.1 | m | y |
| 2 | Pos_050101_namespaces_002 | Verify schema with no target namespace is correctly translated into single module | Clause 5.1.1 | m | y |
| 3 | Pos_050101_namespaces_003 | Verify that two schemas with the same target namespace are correctly translated | Clause 5.1.1 | m | y |
| 4 | Pos_050101_namespaces_004 | Verify that two schemas with no target namespace are correctly translated | Clause 5.1.1 | m | y |

## 5.4. Includes

*Table 3. Includes*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_050102_includes_001 | Test inclusion of a schema with the same namespace | Clause 5.1.2 | m | y |
| 2 | Pos_050102_includes_002 | Verify that included schema with no target namespace is transformed twice (inclusion) | Clause 5.1.2 | m | y |
| 3 | Pos_050102_includes_003 | Verify that included schema with no target namespace is transformed twice (no namespace) | Clause 5.1.2 | m | y |

## 5.5. Imports

*Table 4. Imports*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_050103_imports_001 | Verify that it is not allowed to import imports from XSD schemas | Clause 5.1.3 | m | y |
| 2 | Pos_050103_imports_001 | Verify that XSD import statement is handled correctly | Clause 5.1.3 | m | y |

## 5.6. Attributes of the XSD schema element

*Table 5. Attributes of the XSD schema element*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_050104_attributes_of_the_xsd_schema_element_001 | Verify that qualified default element form is correctly processed (no namespace prefix) | Clause 5.1.4 | m | y |
| 2 | Pos_050104_attributes_of_the_xsd_schema_element_002 | Verify that qualified default element form is correctly processed (namespace prefix used) | Clause 5.1.4 | m | y |
| 3 | Pos_050104_attributes_of_the_xsd_schema_element_003 | Verify that unqualified default element form is correctly processed | Clause 5.1.4 | m | y |
| 4 | Pos_050104_attributes_of_the_xsd_schema_element_004 | Verify that qualified default attribute form is correctly processed (no namespace prefix) | Clause 5.1.4 | m | y |
| 5 | Pos_050104_attributes_of_the_xsd_schema_element_005 | Verify that qualified default attribute form is correctly processed (namespace prefix used) | Clause 5.1.4 | m | y |
| 6 | Pos_050104_attributes_of_the_xsd_schema_element_006 | Verify that unqualified default attribute form is correctly processed | Clause 5.1.4 | m | y |

# 5.7. Name conversion rules

*Table 6. Name conversion rules*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_050202_name_conversion_rules_001 | Verify conversion of symbols into U+005f (low line) | Clause 5.2.2 | m | y |
| 2 | Pos_050202_name_conversion_rules_002 | Verify that non-ASCI letters are not present in transforming identifiers | Clause 5.2.2 | m | y |
| 3 | Pos_050202_name_conversion_rules_003 | Verify that multiple "_" are simplified in transforming identifiers | Clause 5.2.2 | m | y |
| 4 | Pos_050202_name_conversion_rules_004 | Verify that leading and trailing low lines are removed | Clause 5.2.2 | m | y |
| 5 | Pos_050202_name_conversion_rules_005 | Verify that type names are capitalized | Clause 5.2.2 | m | y |
| 6 | Pos_050202_name_conversion_rules_006 | Verify that prefixing type names with "X" works correctly | Clause 5.2.2 | m | y |
| 7 | Pos_050202_name_conversion_rules_007 | Verify that names of field of structure types are uncapitalized | Clause 5.2.2 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 8 | Pos_050202_name_conversion_rules_008 | Verify that names of enumerated items are uncapitalized | Clause 5.2.2 | m | y |
| 9 | Pos_050202_name_conversion_rules_009 | Verify that prefixing field names of structured types with "x" works correctly | Clause 5.2.2 | m | y |
| 10 | Pos_050202_name_conversion_rules_010 | Verify that prefixing enumerated items with "x" works correctly | Clause 5.2.2 | m | y |
| 11 | Pos_050202_name_conversion_rules_011 | Check transformation of empty type identifier into "X" | Clause 5.2.2 | m | y |
| 12 | Pos_050202_name_conversion_rules_012 | Check transformation of empty structured field identifier into "x" | Clause 5.2.2 | m | y |
| 13 | Pos_050202_name_conversion_rules_013 | Check transformation of empty enumerated value into "x" | Clause 5.2.2 | m | y |
| 14 | Pos_050202_name_conversion_rules_014 | Verify that additional suffices are attached in case of name clashes between types | Clause 5.2.2 | m | y |
| 15 | Pos_050202_name_conversion_rules_015 | Verify that suffix is attached in case of name clash between types and local module | Clause 5.2.2 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 16 | Pos_050202_name_conversion_rules_016 | Verify that suffix is attached in case of name clash between types and imported module | Clause 5.2.2 | m | y |
| 17 | Pos_050202_name_conversion_rules_017 | Verify that suffix is attached in case of name clash between field names | Clause 5.2.2 | m | y |
| 18 | Pos_050202_name_conversion_rules_018 | Verify that suffix is attached in case of name clash between field name and keyword | Clause 5.2.2 | m | y |
| 19 | Pos_050202_name_conversion_rules_019 | Verify that suffix is attached in case of name clash between field name and predefined function | Clause 5.2.2 | m | y |
| 20 | Pos_050202_name_conversion_rules_020 | Verify that suffix is attached in case of name clash between enumerated items | Clause 5.2.2 | m | y |
| 21 | Pos_050202_name_conversion_rules_021 | Verify that suffix is attached in case of name clash between enumerated item and keyword | Clause 5.2.2 | m | y |
| 22 | Pos_050202_name_conversion_rules_022 | Verify that suffix is attached in case of name clash between enumerated item and predefined function | Clause 5.2.2 | m | y |
| 23 | Pos_050202_name_conversion_rules_023 | Verify that name clash between module names is resolved using suffix | Clause 5.2.2 | m | y |

## 5.8. Order of the mapping

*Table 7. Order of the mapping*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_050203_order_of_the_mapping_001 | Verify order of top-level schema components | Clause 5.2.3 | m | y |
| 2 | Pos_050203_order_of_the_mapping_002 | Verify that alphabetical sorting is based on character ordinal numbers | Clause 5.2.3 | m | y |
| 3 | Pos_050203_order_of_the_mapping_003 | Verify that alphabetical sorting is done only inside sets of items | Clause 5.2.3 | m | y |
| 4 | Pos_050203_order_of_the_mapping_004 | Assure that namespaces are ordered lexically | Clause 5.2.3 | m | y |
| 5 | Pos_050203_order_of_the_mapping_005 | Assure that namespaces are ordered lexically | Clause 5.2.3 | m | y |

## 5.9. Built-in data types

*Table 8. Built-in data types*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_06_top_level_001 | Verify conversion of simpleType based on built-in XSD type | Clause 6 | m | y |

## 5.10. Length

*Table 9. Length*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060101_length_001 | Verify that a length-restricted XSD type shall be mapped to a corresponding length restricted TTCN-3 type. | Clause 6.1.1 | m | y |
| 2 | Pos_060101_length_001 | Verify that a length-restricted XSD type shall be mapped to a corresponding length restricted TTCN-3 type. | Clause 6.1.1 | m | y |
| 3 | Pos_060101_length_002 | Verify that a length-restricted XSD type shall be mapped to a corresponding length restricted TTCN-3 type. | Clause 6.1.1 | m | y |

## 5.11. Enumeration

*Table 10. Enumeration*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060105_enumeration_001 | Verify if tool rejects validation in case of restricted value due xsd type declaration. | Clause 6.1.5 | m | y |
| 2 | Neg_060105_enumeration_002 | Verify if tool rejects validation in case of restricted enumerated value length due xsd type declaration. | Clause 6.1.5 | m | y |
| 3 | Neg_060105_enumeration_003 | Verify if tool rejects validation in case of restricted value due xsd type declaration. | Clause 6.1.5 | m | y |
| 4 | Neg_060105_enumeration_004 | Disallow enumeration values removed by restriction | Clause 6.1.5 | m | y |
| 5 | Pos_060105_enumeration_001 | Verify mapping of simple type definition that is a restriction of string type with an enumeration facet | Clause 6.1.5 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 6 | Pos_060105_enumeration_002 | Verify mapping of simple type definition that is a restriction of integer type with an enumeration facet | Clause 6.1.5 | m | y |
| 7 | Pos_060105_enumeration_003 | Verify mapping of simple type definition that is a restriction of integer type with a minInclusive and a maxInclusive facet | Clause 6.1.5 | m | y |
| 8 | Pos_060105_enumeration_004 | Verify mapping of simple type definition that is a restriction of another simple type of definition, derived by restriction from integer type with the addition of a minInclusive and a maxInclusive facet | Clause 6.1.5 | m | y |
| 9 | Pos_060105_enumeration_005 | Verify mapping of simple type definition that is a restriction of another type definition, derived by restriction from string with the addition of an enumeration facet | Clause 6.1.5 | m | y |
| 10 | Pos_060105_enumeration_006 | Verify mapping of simple type definition that is a restriction of another simple type definition, derived by restriction from string with the addition of an enumeration facet | Clause 6.1.5 | m | y |

## 5.12. MinInclusive

*Table 11. MinInclusive*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060107_minInclusive_001 | Verify mapping of an integer element with a minInclusive facet | Clause 6.1.7 | m | y |
| 2 | Pos_060107_minInclusive_002 | Verify mapping of a float element with a numeric minInclusive value | Clause 6.1.7 | m | y |
| 3 | Pos_060107_minInclusive_003 | Verify mapping of a float element with special minInclusive values | Clause 6.1.7 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 4 | Pos_060107_mininclusive_004 | Verify mapping of a float element with special minInclusive values | Clause 6.1.7 | m | y |
| 5 | Pos_060107_mininclusive_005 | Verify mapping of a float element with special minInclusive values | Clause 6.1.7 | m | y |

## 5.13. MaxInclusive

*Table 12. MaxInclusive*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060108_maxinclusive_001 | Verify mapping of elements of type integer with maxInclusive facet | Clause 6.1.8 | m | y |
| 2 | Pos_060108_maxinclusive_002 | Verify mapping of a float type with a numeric maxInclusive facet | Clause 6.1.8 | m | y |
| 3 | Pos_060108_maxinclusive_003 | Verify mapping of a float type with a numeric maxInclusive facet | Clause 6.1.8 | m | y |
| 4 | Pos_060108_maxinclusive_004 | Verify mapping of a float type with a numeric maxInclusive facet | Clause 6.1.8 | m | y |

## 5.14. MinExclusive

*Table 13. MinExclusive*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060109_minexclusive_001 | Verify if tool rejects validation in case of restricted value due xsd type declaration. | Clause 6.1.9 | m | y |
| 2 | Neg_060109_minexclusive_002 | Verify if tool rejects validation in case of restricted value due xsd type declaration. | Clause 6.1.9 | m | y |
| 3 | Pos_060109_minexclusive_001 | Verify if tool accepts values restricted by xsd type declaration. | Clause 6.1.9 | m | y |
| 4 | Pos_060109_minexclusive_002 | Verify if tool accepts values restricted by xsd type declaration. | Clause 6.1.9 | m | y |

## 5.15. MaxExclusive

*Table 14. MaxExclusive*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060110_maxexclusive_001 | Verify that INF (negative infinity) or NaN (not-a-number), this type shall not be translated to TTCN-3 | Clause 6.1.10 | m | y |
| 2 | Pos_060110_maxexclusive_001 | Verify mapping of a maxExclusive facet applied to a type, which is derivative of integer | Clause 6.1.10 | m | y |
| 3 | Pos_060110_maxexclusive_002 | Verify mapping of a maxExclusive facet applied to the float type | Clause 6.1.10 | m | y |
| 4 | Pos_060110_maxexclusive_003 | Verify mapping of a maxExclusive facet applied to the float type | Clause 6.1.10 | m | y |

# 5.16. Total digits

*Table 15. Total digits*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060111_total_digits_001 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 2 | Neg_060111_total_digits_002 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 3 | Neg_060111_total_digits_003 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 4 | Neg_060111_total_digits_004 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 5 | Pos_060111_total_digits_001 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 6 | Pos_060111_total_digits_002 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 7 | Pos_060111_total_digits_003 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 8 | Pos_060111_total_digits_004 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |
| 9 | Pos_060111_total_digits_005 | Check that totalDigits are converted to value boundaries | Clause 6.1.11 | m | y |

## 5.17. Fraction digits

*Table 16. Fraction digits*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060112_fraction_digits_001 | Check that floats having same accuracy as fractionDigits are converted correctly | Clause 6.1.12 | m | y |
| 2 | Pos_060112_fraction_digits_002 | Check that floats having higher accuracy than fractionDigits are converted correctly | Clause 6.1.12 | m | y |

## 5.18. Not specifically mapped facets

*Table 17. Not specifically mapped facets*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060113_not_mapped_001 | Handle not mapped facets to transparent | Clause 6.1.13 | m | n |

## 5.19. String

*Table 18. String*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060201_string_001 | Verify mapping of a string type | Clause 6.2.1 | m | y |

## 5.20. Name

*Table 19. Name*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060204_name_001 | Verify mapping of a Name type | Clause 6.2.4 | m | y |

## 5.21. Any URI

*Table 20. Any URI*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060212_any_uri_001 | Verify mapping of an anyURI type | Clause 6.2.12 | m | y |
| 2 | Neg_060212_any_uri_002 | Verify mapping of an anyURI type | Clause 6.2.12 | m | y |
| 3 | Pos_060212_any_uri_001 | Verify mapping of an anyURI type | Clause 6.2.12 | m | y |

## 5.22. Integer

*Table 21. Integer*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060301_integer_001 | Verify that the integer type shall be translated to TTCN-3 as a plain integer | Clause 6.3.1 | m | y |

## 5.23. Positive integer

*Table 22. Positive integer*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060302_positive_integer_001 | Verify that the integer type shall be translated to TTCN-3 as the range-restricted integer | Clause 6.3.2 | m | y |

## 5.24. Non-positive integer

*Table 23. Non-positive integer*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060303_non_positive_integer_001 | Verify that the non positive integer type shall be translated to TTCN-3 as the range-restricted integer | Clause 6.3.3 | m | y |

## 5.25. Negative integer

*Table 24. Negative integer*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060304_negative_integer_001 | Verify that the negative integer type shall be translated to TTCN-3 as the range-restricted integer | Clause 6.3.4 | m | y |

## 5.26. Non-negative integer

*Table 25. Non-negative integer*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060305_non_negative_integer_001 | Verify that the non negative integer type shall be translated to TTCN-3 as the range-restricted integer | Clause 6.3.5 | m | y |

## 5.27. Long

*Table 26. Long*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060306_long_001 | Verify that long type (64bit) shall be translated to TTCN-3 as a plain long | Clause 6.3.6 | m | y |

## 5.28. Unsigned long

*Table 27. Unsigned long*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060307_unsigned_long_001 | Verify that unsigned long type (64bit) shall be translated to TTCN-3 as a plain unsigned long | Clause 6.3.7 | m | y |

## 5.29. Int

*Table 28. Int*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060308_int_001 | Verify that int type (32 bit) shall be translated to TTCN-3 as a plain long as defined in clause 6.3.8 of 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 | Clause 6.3.8 | m | y |

## 5.30. Unsigned int

*Table 29. Unsigned int*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060309_unsigned_int_001 | Verify that unsigned int type (32 bit) shall be translated to TTCN-3 as a plain unsigned long as defined in clause 6.3.9 of 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 | Clause 6.3.9 | m | y |

## 5.31. Short

*Table 30. Short*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060310_short_001 | Verify that short type (16 bit) shall be translated to TTCN-3 as a plain short as defined in clause 6.3.10 of 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 | Clause 6.3.10 | m | y |

## 5.32. Unsigned Short

*Table 31. Unsigned Short*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060311_unsigned_short_001 | Verify that unsigned short type (16 bit) shall be translated to TTCN-3 as a plain unsigned short as defined in clause 6.3.11 of 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 | Clause 6.3.11 | m | y |

## 5.33. Byte

*Table 32. Byte*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060312_byte_001 | Verify that byte type (8 bit) shall be translated to TTCN-3 as a plain byte as defined in clause 6.3.12 of 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 | Clause 6.3.12 | m | y |

## 5.34. Unsigned byte

*Table 33. Unsigned byte*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060313_unsigned_byte_001 | Verify that unsigned byte type (8 bit) shall be translated to TTCN-3 as a plain unsigned byte as defined in clause 6.3.13 of 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 | Clause 6.3.13 | m | y |

## 5.35. Decimal

*Table 34. Decimal*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060401_decimal_001 | Verify that decimal type shall be translated to TTCN-3 as a plain float | Clause 6.4.1 | m | y |

## 5.36. Float

*Table 35. Float*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060402_float_001 | Verify conversion of XSD float type | Clause 6.4.2 | m | y |

## 5.37. Double

*Table 36. Double*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_060403_double_001 | Verify that double type shall be translated to TTCN-3 as an IEEE754double as defined in clause 6.4.3 of 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 | Clause 6.4.3 | m | y |

# 5.38. Date and time

*Table 37. Date and time*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060502_date_and_time_001 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 2 | Neg_060502_date_and_time_002 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 3 | Neg_060502_date_and_time_003 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 4 | Neg_060502_date_and_time_004 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 5 | Pos_060502_date_and_time_001 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 6 | Pos_060502_date_and_time_002 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 7 | Pos_060502_date_and_time_003 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |
| 8 | Pos_060502_date_and_time_004 | Verify that the dateTime type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.2 | m | y |

# 5.39. Date

*Table 38. Date*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060504_date_001 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 2 | Neg_060504_date_002 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 3 | Neg_060504_date_003 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 4 | Neg_060504_date_004 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 5 | Pos_060504_date_001 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 6 | Pos_060504_date_002 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 7 | Pos_060504_date_003 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |
| 8 | Pos_060504_date_004 | Verify that the date type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.4 | m | y |

# 5.40. Gregorian year and month

*Table 39. Gregorian year and month*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060505_gregorian_year_and_month_001 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 2 | Neg_060505_gregorian_year_and_month_002 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |
| 3 | Neg_060505_gregorian_year_and_month_003 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |
| 4 | Neg_060505_gregorian_year_and_month_004 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |
| 5 | Pos_060505_gregorian_year_and_month_001 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |
| 6 | Pos_060505_gregorian_year_and_month_002 | Verify that the gYearMonth type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.5 | m | y |

# 5.41. Gregorian year

*Table 40. Gregorian year*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_060506_gregorian_year_001 | Verify that the gYear type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.6 | m | y |
| 2 | Pos_060506_gregorian_year_001 | Verify that the gYear type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.6 | m | y |
| 3 | Pos_060506_gregorian_year_002 | Verify that the gYear type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.6 | m | y |
| 4 | Pos_060506_gregorian_year_003 | Verify that the gYear allows year 0 | Clause 6.5.6 | m | y |
| 5 | Pos_060506_gregorian_year_004 | Verify that the gYear type shall be translated to TTCN-3 using the pattern-restricted charstring | Clause 6.5.6 | m | y |
| 6 | Pos_060506_gregorian_year_005 | Verify that the gYear accepts negative years | Clause 6.5.6 | m | y |
| 7 | Pos_060506_gregorian_year_006 | Verify that the gYear allows negative year with more than 4 digits | Clause 6.5.6 | m | y |

# 5.42. Boolean type

*Table 41. Boolean type*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0607_boolean_type_001 | Verify that the XSD boolean type shall be mapped to the TTCN-3 boolean type | Clause 6.7 | m | y |
| 2 | Pos_0607_boolean_type_002 | Verify that the XSD boolean type shall be mapped to the TTCN-3 boolean type | Clause 6.7 | m | y |

## 5.43. AnyType and anySimpleType types

*Table 42. AnyType and anySimpleType types*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0608_anytype_and_anysimpletype_types_001 | Verify conversion of anySimpleType | Clause 6.8 | m | y |
| 2 | Pos_0608_anytype_and_anysimpletype_types_002 | Verify conversion of anyType | Clause 6.8 | m | y |

## 5.44. Id

*Table 43. Id*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070101_id_001 | Verify conversion of id attribute of global element | Clause 7.1.1 | m | n |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 2 | Pos_070101_id_002 | verify conversion of id attribute of local element | Clause 7.1.1 | m | n |

# 5.45. MinOccurs and maxOccurs

*Table 44. MinOccurs and maxOccurs*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070104_minoccurs_and_maxoccurs_001 | a list with minOccurs 0 should not be mapped optional in TTCN-3 | Clause 7.1.4 | m | y |
| 2 | Neg_070104_minoccurs_and_maxoccurs_002 | A restricted length list [5, 10] should not allow less than 5 elements | Clause 7.1.4 | m | y |
| 3 | Neg_070104_minoccurs_and_maxoccurs_003 | A restricted length list [5, 10] should not allow more than 10 elements | Clause 7.1.4 | m | y |
| 4 | Pos_070104_minoccurs_and_maxoccurs_001 | Optional field defined by minOccurs has to be mapped as optional in TTCN-3 | Clause 7.1.4 | m | y |
| 5 | Pos_070104_minoccurs_and_maxoccurs_002 | Optional field defined by minOccurs has to exist in TTCN-3 and match the value | Clause 7.1.4 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 6 | Pos_070104_minoccurs_and_maxoccurs_003 | a list with minOccurs 0 should allow zero elements | Clause 7.1.4 | m | y |
| 7 | Pos_070104_minoccurs_and_maxoccurs_004 | A restricted length list (0, unbounded) should allow elements | Clause 7.1.4 | m | y |
| 8 | Pos_070104_minoccurs_and_maxoccurs_005 | A restricted length list [5, 10] should allow 5 elements | Clause 7.1.4 | m | y |
| 9 | Pos_070104_minoccurs_and_maxoccurs_006 | A restricted length list [5, 10] should allow 10 elements | Clause 7.1.4 | m | y |
| 10 | Pos_070104_minoccurs_and_maxoccurs_007 | A restricted length list [5, 10] should allow 7 elements | Clause 7.1.4 | m | y |

# 5.46. Default and Fixed

*Table 45. Default and Fixed*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070105_default_and_fixed_001 | Verify constraint of type based on XSD definition with fixed attribute | Clause 7.1.5 | m | y |
| 2 | Pos_070105_default_and_fixed_001 | Verify conversion of fixed attribute | Clause 7.1.5 | m | y |
| 3 | Pos_070105_default_and_fixed_002 | Verify conversion of default attribute | Clause 7.1.5 | m | y |
| 4 | Pos_070105_default_and_fixed_003 | Verify that default value is automatically assigned to empty element by decoder | Clause 7.1.5 | m | y |
| 5 | Pos_070105_default_and_fixed_004 | Verify that fixed value is automatically assigned to empty element by decoder | Clause 7.1.5 | m | y |

## 5.47. Form

*Table 46. Form*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070106_form_001 | check correct namespace prefix encoding for elementFormDefault | Clause 7.1.6 | m | n |
| 2 | Neg_070106_form_002 | check correct namespace prefix encoding for elementFormDefault | Clause 7.1.6 | m | n |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 3 | Neg_070106_form_003 | check correct namespace prefix encoding for attributeFormDefault | Clause 7.1.6 | m | n |
| 4 | Neg_070106_form_004 | check correct namespace prefix encoding for attributeFormDefault | Clause 7.1.6 | m | n |
| 5 | Pos_070106_form_001 | Verify that unqualified attribute form is correctly converted (unqualified attributeFormDefault) | Clause 7.1.6 | m | y |
| 6 | Pos_070106_form_002 | Verify that unqualified attribute form is correctly converted (qualified attributeFormDefault) | Clause 7.1.6 | m | y |
| 7 | Pos_070106_form_003 | Verify that qualified attribute form is correctly converted (unqualified attributeFormDefault) | Clause 7.1.6 | m | y |
| 8 | Pos_070106_form_004 | Verify that qualified attribute form is correctly converted (qualified attributeFormDefault) | Clause 7.1.6 | m | y |
| 9 | Pos_070106_form_005 | Verify that unqualified element form is correctly converted (unqualified elementFormDefault) | Clause 7.1.6 | m | y |
| 10 | Pos_070106_form_006 | Verify that unqualified element form is correctly converted (qualified elementFormDefault) | Clause 7.1.6 | m | y |
| 11 | Pos_070106_form_007 | Verify that qualified element form is correctly converted (unqualified elementFormDefault) | Clause 7.1.6 | m | y |
| 12 | Pos_070106_form_008 | Verify that qualified element form is correctly converted (qualified elementFormDefault) | Clause 7.1.6 | m | y |
| 13 | Pos_070106_form_009 | check correct namespace prefix encoding for elementFormDefault | Clause 7.1.6 | m | y |
| 14 | Pos_070106_form_010 | check correct namespace prefix encoding for elementFormDefault | Clause 7.1.6 | m | y |
| 15 | Pos_070106_form_011 | check correct namespace prefix encoding for attributeFormDefault | Clause 7.1.6 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 16 | Pos_070106_form_012 | check correct namespace prefix encoding for attributeFormDefault | Clause 7.1.6 | m | y |

## 5.48. Type

*Table 47. Type*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070107_type_001 | Verify conversion of type attribute referencing global simpleType | Clause 7.1.7 | m | y |
| 2 | Pos_070107_type_002 | Verify conversion of type attribute referencing global complexType | Clause 7.1.7 | m | y |
| 3 | Pos_070107_type_003 | Verify conversion of type attribute referencing built-in type | Clause 7.1.7 | m | y |

## 5.49. Use

*Table 48. Use*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070112_use_001 | Verify that attribute with required use cannot be omitted | Clause 7.1.12 | m | y |
| 2 | Pos_070112_use_001 | Verify that attribute with required use is correctly converted | Clause 7.1.12 | m | y |
| 3 | Pos_070112_use_002 | Verify that attribute with optional use is correctly converted | Clause 7.1.12 | m | y |
| 4 | Pos_070112_use_003 | Verify that attribute with prohibited use is not converted | Clause 7.1.12 | m | y |

# 5.50. Final

*Table 49. Final*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070114_final_001 | Verify conversion of elements with final attribute | Clause 7.1.14 | m | y |

# 5.51. Element component

*Table 50. Element component*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0703_element_component_001 | Verify conversion of global element of simple type | Clause 7.3 | m | y |
| 2 | Pos_0703_element_component_002 | Verify conversion of global element of user defined type | Clause 7.3 | m | y |
| 3 | Pos_0703_element_component_003 | Verify conversion of global element of locally defined complex type | Clause 7.3 | m | y |
| 4 | Pos_0703_element_component_004 | Verify conversion of local elements defined by reference with different namespace | Clause 7.3 | m | y |

# 5.52. Attribute element definitions

*Table 51. Attribute element definitions*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070401_attribute_element_definitions_001 | Verify mapping of a globally defined attribute | Clause 7.4.1 | m | y |

## 5.53. Attribute group definitions

*Table 52. Attribute group definitions*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070402_attribute_group_definitions_001 | Verify mapping of a globally defined attribute group | Clause 7.4.2 | m | y |

## 5.54. Derivation by restriction

*Table 53. Derivation by restriction*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070501_derivation_by_restriction_001 | Verify that it is possible to convert anonymously | Clause 7.5.1 | m | y |

## 5.55. Derivation by list

*Table 54. Derivation by list*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070502_derivation_by_list_001 | Verify length constraint imposed on type derived by list | Clause 7.5.2 | m | y |
| 2 | Neg_070502_derivation_by_list_002 | Verify constraint imposed on inner type defined inside XSD list | Clause 7.5.2 | m | y |
| 3 | Pos_070502_derivation_by_list_001 | Verify that derivation by list is converted to record of | Clause 7.5.2 | m | y |
| 4 | Pos_070502_derivation_by_list_002 | Verify mapping of facets connected applied to derivation by list | Clause 7.5.2 | m | y |
| 5 | Pos_070502_derivation_by_list_003 | Verify conversion of facets defined inside XSD list | Clause 7.5.2 | m | y |
| 6 | Pos_070502_derivation_by_list_004 | Verify transformation of derivation by list with enumerated facets inside | Clause 7.5.2 | m | y |
| 7 | Pos_070502_derivation_by_list_005 | Verify transformation of list containing union content | Clause 7.5.2 | m | y |

# 5.56. Derivation by union

*Table 55. Derivation by union*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070503_derivation_by_union_001 | Verify transformation of union with memberTypes attribute | Clause 7.5.3 | m | y |
| 2 | Pos_070503_derivation_by_union_002 | Verify transformation of union with unnamed member types | Clause 7.5.3 | m | y |
| 3 | Pos_070503_derivation_by_union_003 | Verify transformation of union with memberTypes attribute and unnamed member types | Clause 7.5.3 | m | y |
| 4 | Pos_070503_derivation_by_union_004 | Verify transformation of union with memberTypes attribute and unnamed enumeration | Clause 7.5.3 | m | y |
| 5 | Pos_070503_derivation_by_union_005 | Verify transformation of union content containing enumeration facets | Clause 7.5.3 | m | y |
| 6 | Pos_070503_derivation_by_union_006 | Verify transformation of union containing list content | Clause 7.5.3 | m | y |

# 5.57. Extending simple content

*Table 56. Extending simple content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60101_e xtending _simple_ content_ 001 | Verify extension of a built-in type by adding an attribute | Clause 7.6.1.1 | m | y |

# 5.58. Restricting simple content

*Table 57. Restricting simple content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_070 60102_re stricting _simple_ content_ 001 | Verify restriction of a base type | Clause 7.6.1.2 | m | y |
| 2 | Pos_070 60102_re stricting _simple_ content_ 001 | Verify restriction of a base type | Clause 7.6.1.2 | m | y |

# 5.59. Complex content derived by extension

*Table 58. Complex content derived by extension*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60201_d erived_b y_extens ion_001 | Verify mapping of complex type where both the base and the extending types have the compositor sequence | Clause 7.6.2.1 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 2 | Pos_07060201_derived_by_extension_002 | Verify mapping of complex type where both the base and the extending types have the compositor sequence and multiple occurrences are allowed | Clause 7.6.2.1 | m | y |
| 3 | Pos_07060201_derived_by_extension_003 | Verify mapping of complex type where both the base and the extending types have the compositor sequence and multiple occurrences are allowed | Clause 7.6.2.1 | m | y |
| 4 | Pos_07060201_derived_by_extension_004 | Verify mapping of complex type where both the base and the extending types have the compositor sequence and multiple occurrences are allowed | Clause 7.6.2.1 | m | y |
| 5 | Pos_07060201_derived_by_extension_005 | Verify mapping of complex type where both the base and the extending types have the compositor sequence and multiple occurrences are allowed | Clause 7.6.2.1 | m | y |
| 6 | Pos_07060201_derived_by_extension_006 | Verify mapping of complex type where both the base and the extending types have the compositor choice | Clause 7.6.2.1 | m | y |
| 7 | Pos_07060201_derived_by_extension_007 | Verify mapping of complex type where extension of a sequence base type by a choice model group | Clause 7.6.2.1 | m | y |
| 8 | Pos_07060201_derived_by_extension_008 | Verify mapping of complex type: extending of a base type with choice model group by a sequence model group | Clause 7.6.2.1 | m | y |
| 9 | Pos_07060201_derived_by_extension_009 | Verify mapping of complex type: Recursive extension of an anonymous inner type is realized using the TTCN-3 dot notation (starts from the name of the outmost type) | Clause 7.6.2.1 | m | y |

# 5.60. Complex content derived by restriction

*Table 59. Complex content derived by restriction*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060202_derived_by_restriction_001 | Verify mapping of complex content derived by restriction | Clause 7.6.2.2 | m | y |

# 5.61. Referencing group components

*Table 60. Referencing group components*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070603_referencing_group_components_001 | Verify conversion of group reference occurring as child of complex type (sequence, one occurrence) | Clause 7.6.3 | m | y |
| 2 | Pos_070603_referencing_group_components_002 | Verify conversion of group reference occurring inside sequence | Clause 7.6.3 | m | y |
| 3 | Pos_070603_referencing_group_components_003 | Verify conversion of group reference occurring as child of complex type (sequence, optional occurrence) | Clause 7.6.3 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 4 | Pos_070603_referencing_group_components_004 | Verify conversion of group reference occurring as child of complex type (sequence, 0..N) | Clause 7.6.3 | m | y |
| 5 | Pos_070603_referencing_group_components_005 | Verify conversion of group reference occurring as child of complex type (all, one occurrence) | Clause 7.6.3 | m | y |
| 6 | Pos_070603_referencing_group_components_006 | Verify conversion of group reference occurring as child of complex type (all, 0..1) | Clause 7.6.3 | m | y |
| 7 | Pos_070603_referencing_group_components_007 | Verify conversion of group reference occurring as child of complex type (choice, one occurrence) | Clause 7.6.3 | m | y |
| 8 | Pos_070603_referencing_group_components_008 | Verify conversion of group reference occurring as child of complex type (choice, 0..1) | Clause 7.6.3 | m | y |
| 9 | Pos_070603_referencing_group_components_009 | Verify conversion of group reference occurring as child of complex type (choice, 0..N) | Clause 7.6.3 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 10 | Pos_070603_referencing_group_components_010 | Verify conversion of group reference occurring inside choice | Clause 7.6.3 | m | y |

# 5.62. All content

*Table 61. All content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070604_all_content_001 | Verify conversion of all content containing mandatory fields | Clause 7.6.4 | m | y |
| 2 | Pos_070604_all_content_002 | Verify conversion of all content with minOccurs=``0'' | Clause 7.6.4 | m | y |
| 3 | Pos_070604_all_content_003 | Verify transformation of elements with minOccurs attribute occurring inside all content | Clause 7.6.4 | m | y |
| 4 | Pos_070604_all_content_004 | Verify transformation of all content containing attributes | Clause 7.6.4 | m | y |

# 5.63. Choice content

*Table 62. Choice content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070605_top_level_001 | Verify that choice content with minOccurs different than 1 is correctly transformed | Clause 7.6.5 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 2 | Pos_070605_top_level_002 | Verify that choice content with maxOccurs larger than 1 is correctly transformed | Clause 7.6.5 | m | y |

# 5.64. Choice with nested elements

*Table 63. Choice with nested elements*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060501_choice_with_nested_elements_001 | Verify that choice content with nested elements is correctly transformed | Clause 7.6.5.1 | m | y |

# 5.65. Choice with nested group

*Table 64. Choice with nested group*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060502_choice_with_nested_group_001 | Verify that choice content with nested group is correctly transformed | Clause 7.6.5.2 | m | y |

# 5.66. Choice with nested choice

*Table 65. Choice with nested choice*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60503_c hoice_wi th_neste d_choice _001 | Verify that choice content with nested choice is correctly transformed | Clause 7.6.5.3 | m | y |

# 5.67. Choice with nested sequence

*Table 66. Choice with nested sequence*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60504_c hoice_wi th_neste d_seque nce_001 | Verify that choice content with nested sequence is correctly transformed | Clause 7.6.5.4 | m | y |
| 2 | Pos_070 60504_c hoice_wi th_neste d_seque nce_002 | Verify that choice content with multiple nested sequences is correctly transformed | Clause 7.6.5.4 | m | y |

# 5.68. Choice with nested any

*Table 67. Choice with nested any*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60505_c hoice_wi th_neste d_any_0 01 | Verify that choice content with nested any is correctly transformed | Clause 7.6.5.5 | m | y |

## 5.69. Sequence with nested element content

*Table 68. Sequence with nested element content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060601_sequence_with_nested_element_001 | Verify that sequence content with nested elements is correctly transformed | Clause 7.6.6.1 | m | y |

## 5.70. Sequence with nested group content

*Table 69. Sequence with nested group content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060602_sequence_with_nested_group_001 | Verify that sequence content with group reference is correctly transformed | Clause 7.6.6.2 | m | y |

## 5.71. Sequence with nested choice content

*Table 70. Sequence with nested choice content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060603_sequence_with_nested_choice_001 | Verify that sequence content with nested choice is correctly transformed | Clause 7.6.6.3 | m | y |

## 5.72. Sequence with nested sequence content

*Table 71. Sequence with nested sequence content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060604_sequence_with_nested_sequence_001 | Verify that sequence content with sequence is correctly transformed | Clause 7.6.6.4 | m | y |
| 2 | Pos_07060604_sequence_with_nested_sequence_002 | Verify that sequence content with various nested particles is correctly transformed | Clause 7.6.6.4 | m | y |

## 5.73. Sequence with nested any content

*Table 72. Sequence with nested any content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_07060605_sequence_with_nested_any_content_001 | Verify that sequence content with nested any content is correctly transformed | Clause 7.6.6.5 | m | y |

## 5.74. Effect of the minOccurs and maxOccurs attributes on the mapping

*Table 73. Effect of the minOccurs and maxOccurs attributes on the mapping*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070 60606_ef fect_of_ minoccu rs_and_ maxocc urs_001 | Verify that sequences with minOccurs=0 are correctly converted to optional fields | Clause 7.6.6.6 | m | y |
| 2 | Pos_070 60606_ef fect_of_ minoccu rs_and_ maxocc urs_002 | Verify that nested sequences are correctly converted to optional fields | Clause 7.6.6.6 | m | y |
| 3 | Pos_070 60606_ef fect_of_ minoccu rs_and_ maxocc urs_003 | Verify that sequences with minOccurs=unbounded are correctly converted to record of fields | Clause 7.6.6.6 | m | y |
| 4 | Pos_070 60606_ef fect_of_ minoccu rs_and_ maxocc urs_004 | Verify that nested sequences are correctly converted to record of fields | Clause 7.6.6.6 | m | y |

# 5.75. Attribute definitions, attribute and attributeGroup references

*Table 74. Attribute definitions, attribute and attributeGroup references*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070607_attribute_definitions_attribute_and_attributegroup_references_001 | Verify referencing an attributeGroup in a complexType | Clause 7.6.7 | m | y |
| 2 | Pos_070607_attribute_definitions_attribute_and_attributegroup_references_002 | Verify mapping of a local attributes, attribute references and attribute group references without a target namespace | Clause 7.6.7 | m | y |
| 3 | Pos_070607_attribute_definitions_attribute_and_attributegroup_references_003 | Verify mapping of a local attributes, attribute references and attribute group references with a target namespace | Clause 7.6.7 | m | y |

# 5.76. Mixed content

*Table 75. Mixed content*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070608_mixed_content_001 | Verify transformation of complex type with sequence constructor and mixed content type | Clause 7.6.8 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 2 | Pos_070608_mixed_content_002 | Verify transformation of complex type definition with sequence constructor of multiple occurrences and mixed content type | Clause 7.6.8 | m | n |
| 3 | Pos_070608_mixed_content_003 | Verify transformation of complex type definition with all constructor and mixed content type | Clause 7.6.8 | m | y |
| 4 | Pos_070608_mixed_content_004 | Verify transformation of complex type definition with all constructor, optional elements and mixed content type | Clause 7.6.8 | m | n |
| 5 | Pos_070608_mixed_content_005 | Verify transformation of complex type definition with all constructor, optional elements and mixed content type | Clause 7.6.8 | m | y |

# 5.77. The any element

*Table 76. The any element*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070701_the_any_element_001 | Verify conversion of the any element without namespace attribute | Clause 7.7.1 | m | y |
| 2 | Pos_070701_the_any_element_002 | Verify conversion of the any element with ##any namespace | Clause 7.7.1 | m | y |
| 3 | Pos_070701_the_any_element_003 | Verify conversion of the any element with ##local namespace | Clause 7.7.1 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 4 | Pos_070701_the_any_element_004 | Verify conversion of the any element with ##other namespace | Clause 7.7.1 | m | y |
| 5 | Pos_070701_the_any_element_005 | Verify conversion of the any element with ##targetNamespace namespace | Clause 7.7.1 | m | y |
| 6 | Pos_070701_the_any_element_006 | Verify conversion of the any element with URL as namespace into record of | Clause 7.7.1 | m | y |

# 5.78. The anyAttribute element

*Table 77. The anyAttribute element*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_070702_the_anyattribute_element_001 | Verify conversion of anyAttribute element | Clause 7.7.2 | m | y |
| 2 | Pos_070702_the_anyattribute_element_002 | Verify that anyAttribute is converted into optional field | Clause 7.7.2 | m | y |
| 3 | Pos_070702_the_anyattribute_element_003 | Verify that the naming rules apply to converted anyAttribute field | Clause 7.7.2 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 4 | Pos_070702_the_anyattribute_element_004 | Verify that conversion of anyAttribute present both in extended type and extension base | Clause 7.7.2 | m | y |
| 5 | Pos_070702_the_anyattribute_element_005 | Verify that converted anyAttribute field is in correct place | Clause 7.7.2 | m | y |

## 5.79. Annotation

*Table 78. Annotation*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0708_annotation_001 | Verify that XSD annotation can be processed | Clause 7.8 | m | y |

## 5.80. Group components

*Table 79. Group components*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0709_group_components_001 | Verify conversion of group definition with sequence compositor | Clause 7.9 | m | y |
| 2 | Pos_0709_group_components_002 | Verify transformation of group definition with sequence compositor | Clause 7.9 | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 3 | Pos_0709_group_components_003 | Verify conversion of group definition with all compositor | Clause 7.9 | m | y |

# 5.81. Identity-constraint definition schema components

*Table 80. Identity-constraint definition schema components*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_0710_identity_constraint_definition_schema_components_001 | Verify that unique elements (and nested selector and field) are ignored during conversion | Clause 7.10 | m | y |
| 2 | Pos_0710_identity_constraint_definition_schema_components_002 | Verify that key elements (and nested selector and field) are ignored during conversion | Clause 7.10 | m | y |
| 3 | Pos_0710_identity_constraint_definition_schema_components_003 | Verify that keyRef elements (and nested selector and field) are ignored during conversion | Clause 7.10 | m | y |

# 5.82. Head elements of substitution groups

*Table 81. Head elements of substitution groups*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Pos_080101_head_elements_of_substitution_groups_001 | Generic substitution group example | Clause 8.1.1 | m | y |
| 2 | Pos_080101_head_elements_of_substitution_groups_002 | Show effect of the block and abstract attributes on element substitution | Clause 8.1.1 | m | y |
| 3 | Neg_080101_head_elements_of_substitution_groups_002 | Show effect of the block and abstract attributes on element substitution | Clause 8.1.1 | m | y |
| 4 | Pos_080101_head_elements_of_substitution_groups_003 | Blocking substitution | Clause 8.1.1 | m | y |
| 5 | Neg_080101_head_elements_of_substitution_groups_003 | Blocking substitution | Clause 8.1.1 | m | y |

# 5.83. TTCN-3 module XSD

*Table 82. TTCN-3 module XSD*

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 1 | Neg_A_ttcn3_module_xsd_001 | Ensure the builtin XSD type AnySimpleType allows only valid values | Annex A | m | y |
| 2 | Neg_A_ttcn3_module_xsd_002 | Ensure the builtin XSD type AnyType allows only valid values | Annex A | m | y |
| 3 | Neg_A_ttcn3_module_xsd_003 | Ensure the builtin XSD type String allows only valid values | Annex A | m | y |
| 4 | Neg_A_ttcn3_module_xsd_004 | Ensure the builtin XSD type NormalizedString allows only valid values | Annex A | m | y |
| 5 | Neg_A_ttcn3_module_xsd_005 | Ensure the builtin XSD type Token allows only valid values | Annex A | m | y |
| 6 | Neg_A_ttcn3_module_xsd_006 | Ensure the builtin XSD type Name allows only valid values | Annex A | m | y |
| 7 | Neg_A_ttcn3_module_xsd_007 | Ensure the builtin XSD type NMTOKEN allows only valid values | Annex A | m | y |
| 8 | Neg_A_ttcn3_module_xsd_008 | Ensure the builtin XSD type NCName allows only valid values | Annex A | m | y |
| 9 | Neg_A_ttcn3_module_xsd_009 | Ensure the builtin XSD type ID allows only valid values | Annex A | m | y |
| 10 | Neg_A_ttcn3_module_xsd_010 | Ensure the builtin XSD type IDREF allows only valid values | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 11 | Neg_A_ttcn3_module_xsd_011 | Ensure the builtin XSD type ENTITY allows only valid values | Annex A | m | y |
| 12 | Neg_A_ttcn3_module_xsd_012 | Ensure the builtin XSD type HexBinary allows only valid values | Annex A | m | y |
| 13 | Neg_A_ttcn3_module_xsd_013 | Ensure the builtin XSD type Base64Binary allows only valid values | Annex A | m | y |
| 14 | Neg_A_ttcn3_module_xsd_014 | Ensure the builtin XSD type AnyURI allows only valid values | Annex A | m | y |
| 15 | Neg_A_ttcn3_module_xsd_015 | Ensure the builtin XSD type Language allows only valid values | Annex A | m | y |
| 16 | Neg_A_ttcn3_module_xsd_016 | Ensure the builtin XSD type Integer allows only valid values | Annex A | m | y |
| 17 | Neg_A_ttcn3_module_xsd_017 | Ensure the builtin XSD type PositiveInteger allows only valid values | Annex A | m | y |
| 18 | Neg_A_ttcn3_module_xsd_018 | Ensure the builtin XSD type NonPositiveInteger allows only valid values | Annex A | m | y |
| 19 | Neg_A_ttcn3_module_xsd_019 | Ensure the builtin XSD type NegativeInteger allows only valid values | Annex A | m | y |
| 20 | Neg_A_ttcn3_module_xsd_020 | Ensure the builtin XSD type NonNegativeInteger allows only valid values | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 21 | Neg_A_ttcn3_module_xsd_021 | Ensure the builtin XSD type Long allows only valid values | Annex A | m | y |
| 22 | Neg_A_ttcn3_module_xsd_022 | Ensure the builtin XSD type UnsignedLong allows only valid values | Annex A | m | y |
| 23 | Neg_A_ttcn3_module_xsd_023 | Ensure the builtin XSD type Int allows only valid values | Annex A | m | y |
| 24 | Neg_A_ttcn3_module_xsd_024 | Ensure the builtin XSD type UnsignedInt allows only valid values | Annex A | m | y |
| 25 | Neg_A_ttcn3_module_xsd_025 | Ensure the builtin XSD type Short allows only valid values | Annex A | m | y |
| 26 | Neg_A_ttcn3_module_xsd_026 | Ensure the builtin XSD type UnsignedShort allows only valid values | Annex A | m | y |
| 27 | Neg_A_ttcn3_module_xsd_027 | Ensure the builtin XSD type Byte allows only valid values | Annex A | m | y |
| 28 | Neg_A_ttcn3_module_xsd_028 | Ensure the builtin XSD type UnsignedByte allows only valid values | Annex A | m | y |
| 29 | Neg_A_ttcn3_module_xsd_029 | Ensure the builtin XSD type Decimal allows only valid values | Annex A | m | y |
| 30 | Neg_A_ttcn3_module_xsd_030 | Ensure the builtin XSD type Float allows only valid values | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 31 | Neg_A_ttcn3_module_xsd_031 | Ensure the builtin XSD type Double allows only valid values | Annex A | m | y |
| 32 | Neg_A_ttcn3_module_xsd_032 | Ensure the builtin XSD type Duration allows only valid values | Annex A | m | y |
| 33 | Neg_A_ttcn3_module_xsd_033 | Ensure the builtin XSD type DateTime allows only valid values | Annex A | m | y |
| 34 | Neg_A_ttcn3_module_xsd_034 | Ensure the builtin XSD type Time allows only valid values | Annex A | m | y |
| 35 | Neg_A_ttcn3_module_xsd_035 | Ensure the builtin XSD type Date allows only valid values | Annex A | m | y |
| 36 | Neg_A_ttcn3_module_xsd_036 | Ensure the builtin XSD type GYearMonth allows only valid values | Annex A | m | y |
| 37 | Neg_A_ttcn3_module_xsd_037 | Ensure the builtin XSD type GYear allows only valid values | Annex A | m | y |
| 38 | Neg_A_ttcn3_module_xsd_038 | Ensure the builtin XSD type GMonthDay allows only valid values | Annex A | m | y |
| 39 | Neg_A_ttcn3_module_xsd_039 | Ensure the builtin XSD type GDay allows only valid values | Annex A | m | y |
| 40 | Neg_A_ttcn3_module_xsd_040 | Ensure the builtin XSD type GMonth allows only valid values | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 41 | Neg_A_ttcn3_module_xsd_041 | Ensure the builtin XSD type NMTOKENS allows only valid values | Annex A | m | y |
| 42 | Neg_A_ttcn3_module_xsd_042 | Ensure the builtin XSD type IDREFS allows only valid values | Annex A | m | y |
| 43 | Neg_A_ttcn3_module_xsd_043 | Ensure the builtin XSD type ENTITIES allows only valid values | Annex A | m | y |
| 44 | Neg_A_ttcn3_module_xsd_044 | Ensure the builtin XSD type QName allows only valid values | Annex A | m | y |
| 45 | Neg_A_ttcn3_module_xsd_045 | Ensure the builtin XSD type Boolean allows only valid values | Annex A | m | y |
| 46 | Neg_A_ttcn3_module_xsd_046 | Ensure the builtin XSD type XMLCompatibleString allows only valid values | Annex A | m | y |
| 47 | Neg_A_ttcn3_module_xsd_047 | Ensure the builtin XSD type XMLStringWithNoWhitespace allows only valid values | Annex A | m | y |
| 48 | Neg_A_ttcn3_module_xsd_048 | Ensure the builtin XSD type XMLStringWithNoCRLFHT allows only valid values | Annex A | m | y |
| 49 | Pos_A_ttcn3_module_xsd_001 | Ensure the module XSD is available and contains the builtin XSD type AnySimpleType | Annex A | m | y |
| 50 | Pos_A_ttcn3_module_xsd_002 | Ensure the module XSD is available and contains the builtin XSD type AnyType | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 51 | Pos_A_ttcn3_module_xsd_003 | Ensure the module XSD is available and contains the builtin XSD type String | Annex A | m | y |
| 52 | Pos_A_ttcn3_module_xsd_004 | Ensure the module XSD is available and contains the builtin XSD type NormalizedString | Annex A | m | y |
| 53 | Pos_A_ttcn3_module_xsd_005 | Ensure the module XSD is available and contains the builtin XSD type Token | Annex A | m | y |
| 54 | Pos_A_ttcn3_module_xsd_006 | Ensure the module XSD is available and contains the builtin XSD type Name | Annex A | m | y |
| 55 | Pos_A_ttcn3_module_xsd_007 | Ensure the module XSD is available and contains the builtin XSD type NMTOKEN | Annex A | m | y |
| 56 | Pos_A_ttcn3_module_xsd_008 | Ensure the module XSD is available and contains the builtin XSD type NCName | Annex A | m | y |
| 57 | Pos_A_ttcn3_module_xsd_009 | Ensure the module XSD is available and contains the builtin XSD type ID | Annex A | m | y |
| 58 | Pos_A_ttcn3_module_xsd_010 | Ensure the module XSD is available and contains the builtin XSD type IDREF | Annex A | m | y |
| 59 | Pos_A_ttcn3_module_xsd_011 | Ensure the module XSD is available and contains the builtin XSD type ENTITY | Annex A | m | y |
| 60 | Pos_A_ttcn3_module_xsd_012 | Ensure the module XSD is available and contains the builtin XSD type HexBinary | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 61 | Pos_A_ttcn3_module_xsd_013 | Ensure the module XSD is available and contains the builtin XSD type Base64Binary | Annex A | m | y |
| 62 | Pos_A_ttcn3_module_xsd_014 | Ensure the module XSD is available and contains the builtin XSD type AnyURI | Annex A | m | y |
| 63 | Pos_A_ttcn3_module_xsd_015 | Ensure the module XSD is available and contains the builtin XSD type Language | Annex A | m | y |
| 64 | Pos_A_ttcn3_module_xsd_016 | Ensure the module XSD is available and contains the builtin XSD type Integer | Annex A | m | y |
| 65 | Pos_A_ttcn3_module_xsd_017 | Ensure the module XSD is available and contains the builtin XSD type PositiveInteger | Annex A | m | y |
| 66 | Pos_A_ttcn3_module_xsd_018 | Ensure the module XSD is available and contains the builtin XSD type NonPositiveInteger | Annex A | m | y |
| 67 | Pos_A_ttcn3_module_xsd_019 | Ensure the module XSD is available and contains the builtin XSD type NegativeInteger | Annex A | m | y |
| 68 | Pos_A_ttcn3_module_xsd_020 | Ensure the module XSD is available and contains the builtin XSD type NonNegativeInteger | Annex A | m | y |
| 69 | Pos_A_ttcn3_module_xsd_021 | Ensure the module XSD is available and contains the builtin XSD type Long | Annex A | m | y |
| 70 | Pos_A_ttcn3_module_xsd_022 | Ensure the module XSD is available and contains the builtin XSD type UnsignedLong | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 71 | Pos_A_ttcn3_module_xsd_023 | Ensure the module XSD is available and contains the builtin XSD type Int | Annex A | m | y |
| 72 | Pos_A_ttcn3_module_xsd_024 | Ensure the module XSD is available and contains the builtin XSD type UnsignedInt | Annex A | m | y |
| 73 | Pos_A_ttcn3_module_xsd_025 | Ensure the module XSD is available and contains the builtin XSD type Short | Annex A | m | y |
| 74 | Pos_A_ttcn3_module_xsd_026 | Ensure the module XSD is available and contains the builtin XSD type UnsignedShort | Annex A | m | y |
| 75 | Pos_A_ttcn3_module_xsd_027 | Ensure the module XSD is available and contains the builtin XSD type Byte | Annex A | m | y |
| 76 | Pos_A_ttcn3_module_xsd_028 | Ensure the module XSD is available and contains the builtin XSD type UnsignedByte | Annex A | m | y |
| 77 | Pos_A_ttcn3_module_xsd_029 | Ensure the module XSD is available and contains the builtin XSD type Decimal | Annex A | m | y |
| 78 | Pos_A_ttcn3_module_xsd_030 | Ensure the module XSD is available and contains the builtin XSD type Float | Annex A | m | y |
| 79 | Pos_A_ttcn3_module_xsd_031 | Ensure the module XSD is available and contains the builtin XSD type Double | Annex A | m | y |
| 80 | Pos_A_ttcn3_module_xsd_032 | Ensure the module XSD is available and contains the builtin XSD type Duration | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 81 | Pos_A_ttcn3_module_xsd_033 | Ensure the module XSD is available and contains the builtin XSD type DateTime | Annex A | m | y |
| 82 | Pos_A_ttcn3_module_xsd_034 | Ensure the module XSD is available and contains the builtin XSD type Time | Annex A | m | y |
| 83 | Pos_A_ttcn3_module_xsd_035 | Ensure the module XSD is available and contains the builtin XSD type Date | Annex A | m | y |
| 84 | Pos_A_ttcn3_module_xsd_036 | Ensure the module XSD is available and contains the builtin XSD type GYearMonth | Annex A | m | y |
| 85 | Pos_A_ttcn3_module_xsd_037 | Ensure the module XSD is available and contains the builtin XSD type GYear | Annex A | m | y |
| 86 | Pos_A_ttcn3_module_xsd_038 | Ensure the module XSD is available and contains the builtin XSD type GMonthDay | Annex A | m | y |
| 87 | Pos_A_ttcn3_module_xsd_039 | Ensure the module XSD is available and contains the builtin XSD type GDay | Annex A | m | y |
| 88 | Pos_A_ttcn3_module_xsd_040 | Ensure the module XSD is available and contains the builtin XSD type GMonth | Annex A | m | y |
| 89 | Pos_A_ttcn3_module_xsd_041 | Ensure the module XSD is available and contains the builtin XSD type NMTOKENS | Annex A | m | y |
| 90 | Pos_A_ttcn3_module_xsd_042 | Ensure the module XSD is available and contains the builtin XSD type IDREFS | Annex A | m | y |

| Item | TC/TP reference | Purpose | Reference in ETSI ES 201 8739 | Status | Support |
|---|---|---|---|---|---|
| 91 | Pos_A_ttcn3_module_xsd_043 | Ensure the module XSD is available and contains the builtin XSD type ENTITIES | Annex A | m | y |
| 92 | Pos_A_ttcn3_module_xsd_044 | Ensure the module XSD is available and contains the builtin XSD type QName | Annex A | m | y |
| 93 | Pos_A_ttcn3_module_xsd_045 | Ensure the module XSD is available and contains the builtin XSD type Boolean | Annex A | m | y |
| 94 | Pos_A_ttcn3_module_xsd_046 | Ensure the module XSD is available and contains the builtin XSD type XMLCompatibleString | Annex A | m | y |
| 95 | Pos_A_ttcn3_module_xsd_047 | Ensure the module XSD is available and contains the builtin XSD type XMLStringWithNoWhitespace | Annex A | m | y |
| 96 | Pos_A_ttcn3_module_xsd_048 | Ensure the module XSD is available and contains the builtin XSD type XMLStringWithNoCRLFHT | Annex A | m | y |

# Chapter 6. Notes