



---

**Reference**RES/MTS-201873-9 ed471XSD

---

---

**Keywords**language, testing, TTCN-3, XML

---

**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

---

**Important notice**

---

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

---

**Copyright Notification**

---

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2016.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	7
Foreword.....	7
Modal verbs terminology.....	7
1 Scope .....	8
2 References .....	8
2.1 Normative references .....	8
2.2 Informative references.....	9
3 Definitions and abbreviations.....	9
3.1 Definitions.....	9
3.2 Abbreviations .....	10
4 Introduction .....	10
5 Mapping XML Schemas .....	11
5.0 General .....	11
5.1 Namespaces and document references .....	12
5.1.1 Namespaces .....	12
5.1.2 Includes.....	13
5.1.3 Imports.....	14
5.1.4 Attributes of the XSD schema element.....	14
5.1.5 The control namespace .....	15
5.2 Name conversion.....	16
5.2.1 General.....	16
5.2.2 Name conversion rules.....	16
5.2.3 Order of the mapping.....	21
5.3 Mapping of XSD schema components.....	22
5.4 Unsupported features.....	22
5.5 Conformance and compatibility .....	23
6 Built-in data types .....	23
6.0 General .....	23
6.1 Mapping of facets.....	23
6.1.0 General.....	23
6.1.1 Length.....	24
6.1.2 MinLength .....	24
6.1.3 MaxLength.....	25
6.1.4 Pattern .....	25
6.1.5 Enumeration.....	26
6.1.6 WhiteSpace .....	29
6.1.7 MinInclusive .....	29
6.1.8 MaxInclusive .....	31
6.1.9 MinExclusive.....	32
6.1.10 MaxExclusive .....	33
6.1.11 Total digits .....	34
6.1.12 Fraction digits.....	35
6.1.13 Not specifically mapped facets.....	36
6.2 String types.....	36
6.2.0 General.....	36
6.2.1 String .....	37
6.2.2 Normalized string .....	37
6.2.3 Token .....	37
6.2.4 Name.....	37
6.2.5 NMTOKEN .....	37
6.2.6 NCName .....	37
6.2.7 ID.....	37
6.2.8 IDREF.....	38

6.2.9	ENTITY .....	38
6.2.10	Hexadecimal binary .....	38
6.2.11	Base 64 binary .....	38
6.2.12	Any URI .....	38
6.2.13	Language .....	39
6.2.14	NOTATION.....	39
6.3	Integer types .....	39
6.3.0	General.....	39
6.3.1	Integer .....	39
6.3.2	Positive integer .....	39
6.3.3	Non-positive integer .....	39
6.3.4	Negative integer.....	39
6.3.5	Non-negative integer.....	39
6.3.6	Long.....	40
6.3.7	Unsigned long .....	40
6.3.8	Int.....	40
6.3.9	Unsigned int.....	40
6.3.10	Short.....	40
6.3.11	Unsigned Short .....	40
6.3.12	Byte.....	41
6.3.13	Unsigned byte .....	41
6.4	Float types .....	41
6.4.0	General.....	41
6.4.1	Decimal.....	41
6.4.2	Float .....	41
6.4.3	Double .....	41
6.5	Time types .....	41
6.5.0	General.....	41
6.5.1	Duration .....	42
6.5.2	Date and time .....	42
6.5.3	Time.....	42
6.5.4	Date.....	43
6.5.5	Gregorian year and month .....	43
6.5.6	Gregorian year .....	43
6.5.7	Gregorian month and day .....	43
6.5.8	Gregorian day .....	43
6.5.9	Gregorian month.....	43
6.6	Sequence types .....	44
6.6.0	General.....	44
6.6.1	NMTOKENS .....	44
6.6.2	IDREFS .....	44
6.6.3	ENTITIES.....	44
6.6.4	QName.....	44
6.7	Boolean type.....	45
6.8	AnyType and anySimpleType types.....	45
7	Mapping XSD components .....	48
7.0	General .....	48
7.1	Attributes of XSD component declarations.....	48
7.1.0	General.....	48
7.1.1	Id.....	49
7.1.2	Ref .....	49
7.1.3	Name.....	49
7.1.4	MinOccurs and maxOccurs.....	49
7.1.5	Default and Fixed .....	54
7.1.6	Form.....	55
7.1.7	Type.....	55
7.1.8	Mixed.....	56
7.1.9	Abstract.....	56
7.1.10	Block and blockDefault .....	56
7.1.11	Nillable .....	56
7.1.12	Use.....	58

7.1.13	Substitution group.....	58
7.1.14	Final .....	59
7.1.15	Process contents.....	59
7.2	Schema component.....	59
7.3	Element component.....	59
7.4	Attribute and attribute group definitions .....	60
7.4.1	Attribute element definitions .....	60
7.4.2	Attribute group definitions.....	61
7.5	SimpleType components .....	61
7.5.0	General.....	61
7.5.1	Derivation by restriction .....	61
7.5.2	Derivation by list .....	63
7.5.3	Derivation by union .....	64
7.6	ComplexType components.....	67
7.6.0	General.....	67
7.6.1	ComplexType containing simple content .....	68
7.6.1.0	General .....	68
7.6.1.1	Extending simple content .....	68
7.6.1.2	Restricting simple content.....	70
7.6.2	ComplexType containing complex content .....	71
7.6.2.0	General .....	71
7.6.2.1	Complex content derived by extension .....	71
7.6.2.2	Complex content derived by restriction .....	77
7.6.3	Referencing group components .....	79
7.6.4	All content .....	82
7.6.5	Choice content .....	84
7.6.5.0	General .....	84
7.6.5.1	Choice with nested elements .....	84
7.6.5.2	Choice with nested group.....	84
7.6.5.3	Choice with nested choice.....	85
7.6.5.4	Choice with nested sequence.....	86
7.6.5.5	Choice with nested any .....	87
7.6.6	Sequence content .....	87
7.6.6.0	General .....	87
7.6.6.1	Sequence with nested element content.....	87
7.6.6.2	Sequence with nested group content.....	88
7.6.6.3	Sequence with nested choice content .....	88
7.6.6.4	Sequence with nested sequence content.....	89
7.6.6.5	Sequence with nested any content.....	89
7.6.6.6	Effect of the <i>minOccurs</i> and <i>maxOccurs</i> attributes on the mapping .....	90
7.6.7	Attribute definitions, attribute and attributeGroup references .....	92
7.6.8	Mixed content .....	94
7.7	Any and anyAttribute .....	97
7.7.0	General.....	97
7.7.1	The any element.....	97
7.7.2	The anyAttribute element .....	99
7.8	Annotation.....	102
7.9	Group components .....	102
7.10	Identity-constraint definition schema components.....	103
8	Substitutions .....	103
8.0	General .....	103
8.1	Element substitution.....	104
8.1.1	Head elements of substitution groups .....	104
8.1.2	Substitution group members .....	109
8.2	Type substitution .....	109
<b>Annex A (normative):</b>	<b>TTCN-3 module XSD .....</b>	<b>115</b>
<b>Annex B (normative):</b>	<b>Encoding instructions.....</b>	<b>119</b>
B.0	General .....	119

B.1	General .....	119
B.2	The XML encode attribute .....	120
B.3	Encoding instructions .....	120
B.3.1	XSD data type identification .....	120
B.3.2	Any element .....	120
B.3.3	Any attributes .....	121
B.3.4	Attribute .....	122
B.3.5	AttributeFormQualified .....	122
B.3.6	Control namespace identification .....	123
B.3.7	Default for empty .....	123
B.3.8	Element .....	123
B.3.9	ElementFormQualified .....	123
B.3.10	Embed values .....	124
B.3.11	Form .....	124
B.3.12	List .....	125
B.3.13	Name .....	125
B.3.14	Namespace identification .....	126
B.3.15	Nillable elements .....	126
B.3.16	Use union .....	126
B.3.17	Text .....	127
B.3.18	Use number .....	127
B.3.19	Use order .....	128
B.3.20	Whitespace control .....	128
B.3.21	Untagged elements .....	128
B.3.22	Abstract .....	129
B.3.23	Block .....	130
B.3.24	Use type .....	130
B.3.25	Process the content of any elements and attributes .....	131
B.3.26	Transparent .....	131
B.3.27	No Type .....	132
B.3.28	Number of fraction digits .....	132
B.3.29	XML header control .....	133
<b>Annex C (informative):</b>	<b>Examples .....</b>	<b>134</b>
C.0	General .....	134
C.1	Example 1 .....	134
C.2	Example 2 .....	135
C.3	Example 3 .....	137
<b>Annex D (informative):</b>	<b>Deprecated features .....</b>	<b>141</b>
D.1	Using the anyElement encoding instruction to record of fields .....	141
D.2	Using the XML language identifier string .....	141
D.3	Id .....	142
<b>Annex E (informative):</b>	<b>Bibliography .....</b>	<b>143</b>
History .....		144

---

## 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 (<https://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 ETSI Standard (ES) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The present document is part 9 of a multi-part deliverable. Full details of the entire series can be found in part-1 [1].

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

# 1 Scope

The present document defines the mapping rules for W3C Schema (as defined in [7] to [9]) to TTCN-3 as defined in ETSI ES 201 873-1 [1] to enable testing of XML-based systems, interfaces and protocols.

---

## 2 References

### 2.1 Normative 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>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

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

- [1] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".
- [2] 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".
- [3] Recommendation ITU-T X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".
- [4] Recommendation ITU-T X.694: "Information technology - ASN.1 encoding rules: Mapping W3C XML schema definitions into ASN.1".
- [5] World Wide Web Consortium W3C Recommendation: "Extensible Markup Language (XML) 1.1".

NOTE: Available at <http://www.w3.org/TR/xml11>.

- [6] World Wide Web Consortium W3C Recommendation (2006): "Namespaces in XML 1.0".

NOTE: Available at <http://www.w3.org/TR/REC-xml-names/>.

- [7] World Wide Web Consortium W3C Recommendation (2004): "XML Schema Part 0: Primer".

NOTE: Available at <http://www.w3.org/TR/xmlschema-0>.

- [8] World Wide Web Consortium W3C Recommendation (2004): "XML Schema Part 1: Structures".

NOTE: Available at <http://www.w3.org/TR/xmlschema-1>.

- [9] World Wide Web Consortium W3C Recommendation (2004): "XML Schema Part 2: Datatypes".

NOTE: Available at <http://www.w3.org/TR/xmlschema-2>.



## 2.2 Informative 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.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

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] World Wide Web Consortium W3C Recommendation: "SOAP version 1.2, Part 1: Messaging Framework".

NOTE: Available at <http://www.w3.org/TR/soap12>.

[i.2] ISO 8601 (2004): "Data elements and interchange formats - Information interchange - Representation of dates and times".

[i.3] 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.4] Void.

[i.5] Void.

[i.6] Void.

[i.7] Void.

[i.8] ETSI ES 202 789: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Extended TRI".

[i.9] ISO/IEC 10646 (2012): "Information technology - Universal Coded Character Set (UCS)".

[i.10] ISO/IEC 646: "Information technology - ISO 7-bit coded character set for information interchange".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI ES 201 873-1 [1], Recommendation ITU-T X.694 [4] and the following apply:

**alphabetical order:** way of sorting the XSD names based on the code positions of their characters according to ISO/IEC 10646 [i.9]

NOTE: During this sorting the group, plane, row and cell octets is considered, in this order. Names, starting with a character with a smaller code position take precedence. Among the names with identical first character, names containing no more characters take precedence over all other names. Otherwise, names with the second character of smaller code position take precedence, etc. This algorithm is to be continued recursively until all names are sorted into a sequential order.

**schema component:** generic XSD term for the building blocks that comprise the abstract data model of the schema

NOTE: The primary components, which may (type definitions) or obliged to (element and attribute declarations) have names are as follows: simple type definitions, complex type definitions, attribute declarations and element declarations. The secondary components, which are obliged to have names, are as follows: attribute group definitions, identity-constraint definitions, model group definitions and notation declarations. Finally, the "helper" components provide small parts of other components; they are not independent of their context: annotations, model groups, particles, wildcards and attribute uses.

**schema document:** contains a collection of schema components, assembled in a *schema* element information item

NOTE: The target namespace of the schema document may be defined (specified by the *targetNamespace* attribute of the *schema* element) or may be absent (identified by a missing *targetNamespace* attribute of the *schema* element). The latter case is handled in the present document as a particular case of the target namespace being defined.

**target TTCN-3 module:** TTCN-3 module, generated during the conversion, to which the TTCN-3 definition produced by the translation of a given XSD declaration or definition is added

**XML Schema:** represented by a set of schema documents forming a complete specification (i.e. all definitions and references are completely defined)

NOTE: The set may be composed of one or more schema documents, and in the latter case identifying one or more target namespaces (including absence of the target namespace) and more than one schema documents of the set may have the same target namespace (including absence of the target namespace).

**xsi: attributes:** stipulating the content of schema-instances (schema-valid XML documents), XSD defines several attributes for direct use in any XML documents

NOTE: These attributes are in the namespace <http://www.w3.org/2001/XMLSchema-instance>. By convention these XML attributes are referred to by using the prefix "xsi: ", though in practice, any prefix can be used.

## 3.2 Abbreviations

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

ASN.1	Abstract Syntax Notation One
DTD	Document Type Description
SOAP	Simple Object Access Protocol
SUT	System Under Test
TTCN-3	Testing and Test Control Notation version 3
URI	Uniform Resource Identifier
UTF-8	Unicode Transformation Format-8
W3C	World Wide Web Consortium
XER	XML Encoding Rules
XML	eXtensible Markup Language
XSD	XML Schema Definition

---

## 4 Introduction

An increasing number of distributed applications use the XML format to exchange data for various purposes like data bases queries or updates or event telecommunications operations such as provisioning. All of these data exchanges follow very precise rules for data format description in the form of Document Type Description (DTD) [5] and [6] or more recently the proposed XML Schemas [7], [5] and [9]. There are even some XML based communication protocols like SOAP [i.1] that are based on XML Schemas. Like any other communication-based systems, components and protocols, XML based systems, components and protocols are candidates for testing using TTCN-3 [1]. Consequently, there is a need for establishing a mapping between XML data description techniques like DTD or Schemas to TTCN-3 standard data types.

The core language of TTCN-3 is defined in ETSI ES 201 873-1 [1] and provides a full text-based syntax, static semantics and operational semantics as well as a definition for the use of the language with ASN.1 in ETSI ES 201 873-7 [2]. The XML mapping provides a definition for the use of the core language with XML Schema structures and types, enabling integration of XML data with the language as shown in figure 1.

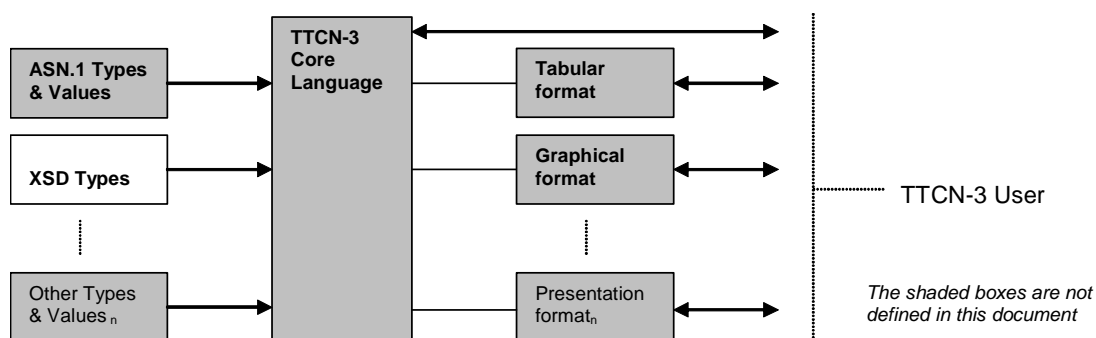


Figure 1: User's view of the core language and the various presentation formats

For compatibility reasons, it is the purpose of the present document that the TTCN-3 code obtained from the XML Schema using the explicit mapping will be the same as the TTCN-3 code obtained from first converting the XML Schema using Recommendation ITU-T X.694 [4] into ASN.1 [3] and then converting the resulting ASN.1 code into TTCN-3 according to ETSI ES 201 873-7 [2]. Moreover, the XML document produced from the TTCN-3 code containing the encoding instructions obtained from the XML Schema based on the present document, will be the same as the XML document produced by the ASN.1 E-XER encoding, when the same XML Schema is converted using Recommendation ITU-T X.694 [4] and the resulted ASN.1 specification is encoded using the E-XER encoding. However, due to the specifics of testing, in a few cases the present document will produce a superset of what Recommendation ITU-T X.694 [4] would produce. For example, according to Recommendation ITU-T X.694 [4], abstract elements are omitted when converting the head element of a substitution group, while the present document includes also the abstract elements into the resulted `union` type, thus allowing provoking the SUT with incorrect data.

## 5 Mapping XML Schemas

### 5.0 General

There are two approaches to the integration of XML Schema and TTCN-3, which will be referred to as implicit and explicit mapping. The implicit mapping makes use of the import mechanism of TTCN-3, denoted by the keywords *language* and *import*. It facilitates the immediate use of data specified in other languages. Therefore, the definition of a specific data interface for each of these languages is required. The explicit mapping translates XML Schema definitions directly into appropriate TTCN-3 language artefacts.

In case of an implicit mapping an internal representation shall be produced from the XML Schema, which representation shall retain all the structural and encoding information. This internal representation is typically not accessible by the user. To make the internal representations related to a given target namespace referenceable in a TTCN-3 module, the module shall explicitly import the target namespace, using its TTCN-3 name equivalent resulting from applying clause 5.2.2 to the namespace. The TTCN-3 import statement shall use the language identifier string specified below. TTCN-3 data types described in clause 5.5 (equivalents to built-in XSD types), in case of an implicit conversion, are internal to the tool and can be referenced in TTCN-3 modules importing any target namespaces of an XSD document explicitly. These types can be also referenced in TTCN-3 modules that explicitly import the XSD module (see annex A). In this case, the import clause refers to the tool's internal representation of the XSD data types and not to an existing module. When importing from an XSD Schema using implicit mapping, the following language identifier string shall be used:

- "XSD".

For explicit mapping, the information present in the XML Schema shall be mapped into accessible TTCN-3 code and - the XML structural information which does not have its correspondent in TTCN-3 code - into accessible encoding instructions. In case of an explicit conversion the TTCN-3 data types described in clause 5.5 (equivalents to built-in XSD types) are not visible in TTCN-3 by default, the user shall import the XSD module (see annex A) explicitly, in addition to the TTCN-3 modules resulted from the conversion. When importing TTCN-3 modules generated by explicit conversion, the use of the "XSD" language clause is optional, but if used, the imported TTCN-3 module shall be appended with one of the XML encode attributes, specified in clause B.2.

The mapping shall start on a set of valid XSD *schema*-s and shall result in a set of valid TTCN-3 modules.

All XSD definitions are **public** by default (see clause 8.2.3 of ETSI ES 201 873-1 [1]).

The examples of the present document are written in the assumption of explicit mapping, although the difference is mainly in accessibility and visibility of generated TTCN-3 code and encoding instruction set.

The present document is structured in three distinct parts:

- Clause 5.5 "Built-in data types" defines the TTCN-3 mapping for all basic XSD data types like strings (see clause 6.2), integers (see clause 6.3), floats (see clause 6.4), etc. and facets (see clause 6.1) that allow for a simple modification of types by restriction of their properties (e.g. restricting the length of a string or the range of an integer).
- Clause 7 "Mapping XSD components" covers the translation of more complex structures that are formed using the components shown in table 1 and a set of XSD attributes (see clause 7.1) which allow for modification of constraints of the resulting types.
- Clause 8 "Substitution" covers the translation of more XSD elements and types that may be substituted for other XSD elements or types respectively in instance documents.

**Table 1: Overview of XSD constructs**

<b>Element</b>	Defines tags that can appear in a conforming XML document.
<b>attribute</b>	Defines attributes for element tags in a conforming XML document.
<b>simpleType</b>	Defines the simplest types. They may be a built-in type, a list or choice of built-in types and they are not allowed to have attributes.
<b>complexType</b>	Defines types that are allowed to be composed, e.g. have attributes and an internal structure.
<b>named model group</b>	Defines a named group of elements.
<b>attribute group</b>	Defines a group of attributes that can be used as a whole in definitions of complexTypes.
<b>identity constraint</b>	Defines that a component has to exhibit certain properties in regard to uniqueness and referencing.

## 5.1 Namespaces and document references

### 5.1.1 Namespaces

A single XML Schema may be composed of a single or several *schema* element information items, and shall be translated to one or more TTCN-3 modules, corresponding to *schema* components that have the same target namespace, including no target namespace. For XSD *schemas* with the same target namespace (including absence of the target namespace) exactly one TTCN-3 module shall be generated.

The names of the TTCN-3 modules generated based on this clause shall be the result of applying the name transformation rules in clause 5.2.2 to the related target namespace, if it exists, or to the predefined name "NoTargetNamespace".

NOTE 1: More than one *schema* element information items in an XML Schema may have the same target namespace, including the case of no target namespace.

The information about the target namespaces and prefixes from the *targetNamespace* and *xmlns* attributes of the corresponding *schema* elements, if exist, shall be preserved in the encoding instruction "namespace as..." attached to the TTCN-3 module. If the target namespace is absent, no "namespace as ..." encoding instruction shall be attached to the TTCN-3 module. All declarations in the module shall inherit the target namespace of the module (including absence of the target namespace).

NOTE 2: If different *schema* element information items using the same target namespace associates different prefixes to that namespace, it is a tool implementation option, which prefix is preserved in the "namespace as..." encoding instruction.

EXAMPLE: Schemas with the same namespace:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://www.example.org"
  targetNamespace="http://www.example.org">
  <!-- makes no difference if this schema is including the next one -->
  :
</xsd:schema>

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://www.example.org"
  targetNamespace="http://www.example.org">
  <!-- makes no difference if this schema is including the previous one -->
  :
</xsd:schema>
```

Will result e.g. in the following TTCN-3 module:

```
module http_www_example_org {
  : // the content of the module is coming from both schemas
}
with {
  encode "XML";
  variant "namespace as 'http://www.example.org' prefix 'ns1'";
  // the prefix in the encoding instruction could also be 'ns2', this is a tool's option.
}
```

## 5.1.2 Includes

XSD *include* element information items shall be ignored if the included *schema* element has the same target namespace as the including one (implying the absence of the target namespace). If the included *schema* element has no target namespace but the including *schema* has (i.e. it is not absent), all definitions of the included *schema* shall be mapped twice, i.e. the resulted TTCN-3 definitions shall be inserted to the TTCN-3 module generated for the *schema* element(s) with no target namespace as well as to the module generated for the *schema* element(s) with the target namespace of the including *schema*.

EXAMPLE: A schema with a target namespace is including a schema without a target namespace:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns="http://www.example.org"
  targetNamespace="http://www.example.org">
  <!-- the including schema -->
  <xsd:include schemaLocation="included.xsd"/>
  :
</xsd:schema>

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--this is the included schema -->
  :
</xsd:schema>
```

Will result the TTCN-3 modules (please note, the content of the modules may come from more than one schemas).

```
module http_www_example_org {
  : // contains definitions mapped from both schemas
}
```