Final draft ETSI ES 201 873-7 V4.9.1 (2021-03)

Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 7: Using ASN.1 with TTCN-3

Reference
RES/MTS-201873-7v491

Keywords
ASN.1, language, testing, TTCN, 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 prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This final draft ETSI Standard (ES) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 7 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 a normative way of using ASN.1 as defined in Recommendations ITU-T X.680 [2], X.681 [3], X.682 [4] and X.683 [5] with TTCN-3. The harmonization of other languages with TTCN-3 is not covered by the present document.

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 https://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] Recommendation ITU-T X.680 (2008): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[3] Recommendation ITU-T X.681 (2008): "Information technology - Abstract Syntax Notation One (ASN.1): Information object specification".

[4] Recommendation ITU-T X.682 (2008): "Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification".

[5] Recommendation ITU-T X.683 (2008): "Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications".

[6] Recommendation ITU-T X.690 (2008): "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[7] Recommendation ITU-T X.691 (2008): "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)".

[8] Recommendation ITU-T X.693 (2008): "Information technology - ASN.1 encoding rules: XML Encoding Rules (XER)".

[9] Recommendation ITU-T T.100 (1988): "International information exchange for interactive Videotex".

[10] Recommendation ITU-T T.101 (1994): "International interworking for Videotex services".

[11] Recommendation ITU-T X.660 (2011): "Information technology - Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree".

[12] 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)".

[13] Recommendation ITU-T X.696 (2015): "Information technology - ASN.1 encoding rules: Specification of Octet Encoding Rules (OER)".

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

[i.2] ETSI ES 201 873-10: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 10: TTCN-3 Documentation Comment Specification".

[i.3] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".

[i.4] Recommendation ITU-T X.121: "Public data networks - Network aspects - International numbering plan for public data networks".

NOTE: References to Recommendations ITU-T include the Recommendation and all Amendments and Corrigenda published to the Recommendation except when specified otherwise in other parts of the present document.

[i.5] Recommendation ITU-T X.208: "Specification of Abstract Syntax Notation One (ASN.1)" (Blue Book).

[i.6] Recommendation ITU-T X.680 (1994): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[i.7] Recommendation ITU-T X.680 (1997): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[i.8] Recommendation ITU-T X.680 (2002): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[i.9] Recommendation ITU-T X.681 (1994): "Information technology - Abstract Syntax Notation One (ASN.1): Information object specification".

[i.10] Recommendation ITU-T X.681 (1997): "Information technology - Abstract Syntax Notation One (ASN.1): Information object specification".

[i.11] Recommendation ITU-T X.681 (2002): "Information technology - Abstract Syntax Notation One (ASN.1): Information object specification".

[i.12] Recommendation ITU-T X.682 (1994): "Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification".

[i.13] Recommendation ITU-T X.682 (1997): "Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification".

[i.14] Recommendation ITU-T X.682 (2002): "Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification".

[i.15] Recommendation ITU-T X.683 (1994): "Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications".

[i.16] Recommendation ITU-T X.683 (1997): "Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications".

[i.17] Recommendation ITU-T X.683 (2002): "Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications".

[i.18] Recommendation ITU-T X.690 (2002): "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[i.19] Recommendation ITU-T X.691 (2002): "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)".

[i.20] 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.21] ETSI ES 202 782: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: TTCN-3 Performance and Real Time Testing".

[i.22] ETSI ES 202 784: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization".

[i.23] ETSI ES 202 785: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types".

[i.24] ETSI ES 202 786: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Support of interfaces with continuous signals".

[i.25] 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.26] CCITT Blue Book.

[i.27] ETSI ES 201 873-8: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 8: The IDL to TTCN-3 Mapping".

[i.28] ETSI ES 201 873-9: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3".

[i.29] ETSI ES 201 873-11: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 11: Using JSON with TTCN-3".

[i.30] ETSI ES 203 022: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language extension: Advanced Matching".

[i.31] ETSI ES 203 790: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Object-Oriented Features".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI ES 201 873-1 [1], Recommendation ITU-T X.660 [11] and the following apply:

associated TTCN-3 type: TTCN-3 type equivalent, obtained by transforming of the corresponding ASN.1 type definition according to clause 9.1 of the present document

NOTE: Associated TTCN-3 types and values may not exist in a visible way; this term is used to identify the part of the abstract information carried by the related ASN.1 type or value, which have significance from the point of view of TTCN-3 (also called the TTCN-3 view).

metatype "OPEN TYPE": used to explain the ASN.1 to TTCN-3 conversion process

NOTE: It does not exist in the input ASN.1 module or the output TTCN-3 module.

root type: Definition in ETSI ES 201 873-1 [1] applies with the following addition: in case of types based on imported ASN.1 types, the root type is determined from the associated TTCN-3 type (see clause 8).

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI ES 201 873-1 [1] and the following apply:

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules of ASN.1
CER	Canonical Encoding Rules of ASN.1
OER	Octet Encoding Rules of ASN.1
OID	Object IDentifier
PER	Packed Encoding Rules of ASN.1
XER	XML Encoding Rules of ASN.1

4 Introduction

When using ASN.1 with TTCN-3 all features of TTCN-3 and statements given in clause 4 of ETSI ES 201 873-1 [1] do apply. In addition, when supporting the present document, TTCN-3 becomes fully harmonized with ASN.1 which may be used with TTCN-3 modules as an alternative data type and value syntax. The present document defines the capabilities required in addition of those specified in ETSI ES 201 873-1 [1] when ASN.1 is supported. The approach used to combine ASN.1 and TTCN-3 could be applied to support the use of other type and value systems with TTCN-3. However, the details of this are not defined in the present document.

ETSI ES 201 873-1 [1] specifies the core capabilities of the TTCN-3 language. Other language mappings (see [i.27], [i.28] and [i.29]) and TTCN-3 language packages as ETSI ES 202 781 [i.20], ETSI ES 202 782 [i.21], ETSI ES 202 784 [i.22], ETSI ES 202 785 [i.23], ETSI ES 202 786 [i.24], ETSI ES 202 789 [i.25], ETSI ES 203 022 [i.30], ETSI ES 203 790 [i.31] may specify extensions to the core language that may define additional rules for the ASN.1 to TTCN-3 mapping. These additional rules are specified in the relevant other documents and need to be supported only if the implementation claims to support the other document.

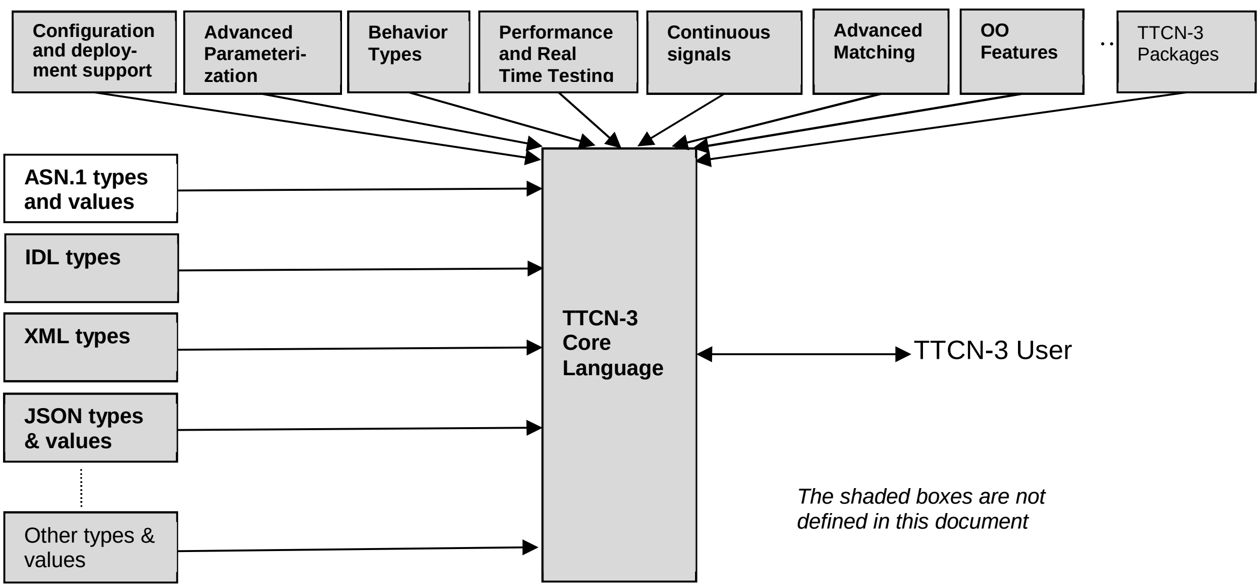

Figure 1: User's view of the core language and its packages

5 General

5.1 Approach

TTCN-3 provides a clean interface for using ASN.1 definitions (as specified in Recommendations ITU-T X.680 [2], X.681 [3], X.682 [4] and X.683 [5]) in TTCN-3 modules.

In general, there are two approaches to the integration of other languages with 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*, in which case the TTCN-3 tool shall produce an internal representation of the imported objects, which representation shall retain all the structural and encoding information. This internal representation is not accessible by the user. It facilitates the immediate use of the abstract data specified in the other language. Therefore, the definition of a specific data interface for each of these languages is required.

The explicit mapping translates the definitions of the other language directly into appropriate TTCN-3 language artefacts. This also means that all information needed for correct encoding and decoding shall be present in the TTCN-3 module(s) generated by this translation.

In case of the ASN.1 to TTCN-3 mapping no TTCN-3 encoding instructions are defined by the present document, hence only the implicit mapping is specified.

5.2 Conformance and compatibility

For an implementation claiming to support the use of ASN.1 with TTCN-3, all features specified in the present document will need to be implemented consistently with the requirements given in the present document and in ETSI ES 201 873-1 [1].

The language mapping presented in the present document is compatible to:

- ETSI ES 201 873-1 [1], V4.9.1.
- ETSI ES 201 873-10 [i.2], V4.5.1.

NOTE: Only the informative annex E uses features from ETSI ES 201 873-10 [i.2].

If later versions of those parts are available and should be used instead, the compatibility of the language mapping presented in the present document has to be checked individually.

6 Amendments to the core language

Using ASN.1 with TTCN-3 is handled at the static type-value level. Though it mainly means additions described in the subsequent clauses, some of the core language syntactical structures shall also be amended to support the use of ASN.1. These are specified in clause A.2.

7 Additional TTCN-3 types

7.1 General

The TTCN-3 types summarized in table 1 shall be supported in addition to those specified in clause 6 of ETSI ES 201 873-1 [1].

Table 1: Overview of TTCN-3 types

Class of type	Keyword	Sub-type
Simple basic types	`objid`	list, range

7.2 The object identifier type

7.2.0 The **objid** type

The object identifier type shall be supported as follows:

- **objid:** a type whose distinguished values are the set of all syntactically correct object identifier values. The value notations for the objid type shall conform to clause 31 of Recommendation ITU-T X.680 [2] with the exception that hyphens are replaced with underscores.

NOTE 1: This definition also allows object identifier values outside the collection of values defined in Recommendation ITU-T X.660 [11] (e.g. with a node beneath the root not defined in Recommendation ITU-T X.660 [11]).

The name form of object identifier components shall be used only for components defined in Recommendation ITU-T X.660 [11]. These predefined object identifier components are given in annex C for information. In case of any conflict between Recommendation ITU-T X.660 [11] and annex C of the present document, the former shall take precedence.

In cases when the identifier of a value referenced within an object identifier value notation is identical to any of the predefined component names, i.e. independently of the position of the predefined component or the referenced value inside the notation (considering name conversion rules in clause 8.2), the name of the referenced value shall be prefixed with the name of the module in which it is defined (see definition of ASN.1 modules in clause 12 of Recommendation ITU-T X.680 [2] and TTCN-3 modules in clause 8.1 of the core language standard ETSI ES 201 873-1 [1]). The prefix and the identifier shall be separated by a dot (.). Predefined object identifier component names may also be prefixed with the name "X660".

NOTE 2: To increase readability it is recommended to use the "X660" prefix also in object identifier values referring to a value identifier that is clashing with any of the predefined component names.

NOTE 3: Rules to resolve name clashes caused by imports are defined in clause 8.2.3.1 of the core language standard ETSI ES 201 873-1 [1].

EXAMPLE:

```
objid{itu_t(0) identified_organization(4) etsi(0)}
// or alternatively
objid {itu_t identified_organization etsi(0)}
// or alternatively
objid { 0 4 0}

// or alternatively
const integer etsi := 0;
const objid itu_idOrg := objid{ itu_t identified_organization }
objid{ itu_idOrg etsi } // note, that both names are referencing value definitions

const integer x := 162;
objid{ itu_t recommendation x A.x }         // it is mandatory to use the module name ('A')
                                            // to prefix the ambiguous identifier
                                            // or alternatively
objid{ itu_t recommendation X660.x A.x }    // the module name shall be present even if
                                            // the "X660" prefix is used
```

7.2.1 Sub-typing of the **objid** type

7.2.1.1 Subtrees of the **objid** type

The object identifier type is a collection of principally infinite set of unique identifier values, each containing a sequence of components; each given sequence of arbitrary length compose an object identifier node as shown on figure 2 (see also annex C). Thus, each node of the object identifier tree - except being a unique identifier itself - is the root of a subtree, containing a potentially infinite number of unique identifiers. The first n components of all the identifiers in the subtree are identical to the components of the node, being the root of the subtree, where n is the number of components of that node. Hence, each object identifier node distinguishes also a unique subset (subtype) of the **objid** type. Each member of this subtype (the subtree) is longer than the node identifying the subtree.

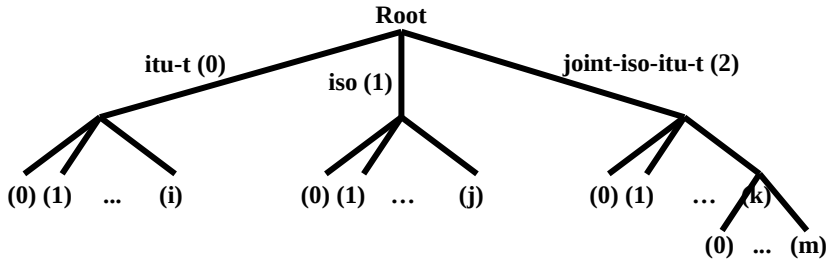

Figure 2: The object identifier tree

Note that object identifiers may also be relative identifiers, i.e. when the given objid value contains only the additional components related to a defined base node. However, the above remains true for relative object identifiers as well, as they also do denote a unique node in the object identifier tree (the node defined by the base node + the relative identifier). If a node is identified by a relative object identifier, all nodes in its subtree will have relative identifiers too.

7.2.1.2 List subtypes

In addition to the types listed in clause 6, table 3 of ETSI ES 201 873-1 [1], value list subtyping of the **objid** type shall be supported. For value lists of the **objid** type the rules in this clause apply. The **objid** nodes in the list shall be of **objid** type and shall be a true subset of the values defined by the **objid** type or the base type being restricted. Clause 7.2.1.1 shall govern in determining if the nodes are a true subset. The subtype defined by this list restricts the allowed values to the concrete nodes on the list and the subtrees identified by them.

EXAMPLE:

```
//identifies the nodes {0 4 0 0}, {0 4 0 1} and all other nodes beneath them
type objid MyObjids (objid{0 4 0 0}, objid{0 4 0 1});

//Further restricting the base type MyObjids
type MyObjids MyNarrowerObjids (objid{0 4 0 0 1 0}, objid{0 4 1 1}, objid{0 4 1 3});

//invalid definition as the node {0 4 2} is not member of the type MyObjids
type MyObjids MyNarrowestObjids (objid{0 4 2 1});
```


7.2.1.3 Range subtypes

In addition to the types listed in clause 6, table 3 of ETSI ES 201 873-1 [1], range subtyping of the **objid** type shall be supported. For range subtyping of the **objid** type the rules in this clause apply. The **objid** nodes determining the lower and the upper bounds of the subtype shall be of **objid** type of equal length and shall be a true subset of the values defined by the **objid** type or the base type being restricted. Clause 7.2.1.1 shall govern in determining if the nodes are a true subset. The subtype defined by the range restricts the allowed values to the nodes between the lower and the upper bounds inclusive and the subtrees identified by them.

EXAMPLE:

```
//identifies the nodes {0 4 0 0}, {0 4 0 1} … {0 4 0 5} and all other nodes beneath them
type objid MyObjidRange (objid{0 4 0 0} .. objid{0 4 0 5});
```

7.2.1.4 Mixing list and range subtypings

It is allowed to mix the list and the range subtyping mechanisms for **objid** types. The nodes identified by the different subtyping mechanisms shall not overlap.

7.2.2 Object identifier values

When defining **objid** values, rules in clauses 5.4.1.1, 8.2.1, 10 and 11.1 of ETSI ES 201 873-1 [1] and in this clause shall apply. In case of inconsistency the present document takes precedence.

Each object identifier node is an object identifier value. In this case the value identifies the concrete node (i.e. with a definite number of components) and does not denote the **objid** subtree beneath it (see clause 7.2.1.1).

7.2.3 Using **objid** values to identify modules

7.2.3.1 Identifying module definitions

When ASN.1 is supported, module names (of the form of a TTCN-3 identifier) may optionally be followed by an object identifier, which shall be a valid value as defined in Recommendation ITU-T X.660 [11].

NOTE: Module names in a test suite may differ in the object identifier part only. However, in this case, due precaution has to be exercised at import to avoid name clash, as prefixing of TTCN-3 identifiers (see clause 8.2.3.1 of ETSI ES 201 873-1 [1]) is unable to resolve such kind of clashes.

7.2.3.2 Identifying modules in import statements

When ASN.1 is supported, in addition to the module names, their object identifiers may also be provided in TTCN-3 import statements. If an object identifier is used as part of the module identifier, this object identifier shall be used by TTCN-3 test systems to identify the correct module.

7.2.4 Object identifier templates

7.2.4.0 General

When defining templates of **objid** types, rules in clause 15 and annex B of ETSI ES 201 873-1 [1] and in this clause shall apply. In case of inconsistency the present document takes precedence.

7.2.4.1 In-line templates

The type of **objid** values can be identified from the value notation alone, hence in addition to the types listed in note 2 of clause 15.4 in ETSI ES 201 873-1 [1], the type specification may also be omitted in case of objid values.

7.2.4.2 Template matching mechanisms

Applicability of matching mechanisms to templates of **objid** types is defined in table 2.

Table 2: TTCN-3 Matching Mechanisms

Used with values of	Value	Instead of values									Inside values			Attributes	
	SpecificValue	OmitValue	ComplementedList	ValueList	AnyValue (?)	AnyValueOrNone (*)	Range	Superset	Subtype	Pattern	AnyElement (?)	AnyElementsOrNone (*)	Permutation	LengthRestriction	IfPresent
objid	Yes	Yes (see note)	Yes	Yes	Yes	Yes (see note)	Yes				Yes	Yes		Yes	Yes (see note)
NOTE: Can be assigned to templates, however when used, shall be applied to optional fields of record and set types only (without restriction on the type of that field).															

The matching mechanisms SpecificValue, OmitValue, AnyValue, AnyValueOrNone and IfPresent are applicable to **objid** fields as well according to the rules defined in ETSI ES 201 873-1 [1].

The value list and complemented value list matching mechanisms can also be used for **objid** templates and template fields. Rules in clauses B.1.2.1 and B.1.2.2 of ETSI ES 201 873-1 [1] also shall apply to **objid** templates.

NOTE: This also means that only the concrete node values on the list is to be considered but not the subtrees identified by them. I.e. in case of a complemented list, a node within a given subtree will match even if the node being the root of the subtree is on the list.

The value range matching mechanism, in addition to types listed in clause B.1.2.5 of ETSI ES 201 873-1 [1], can also be used for **objid** templates. When applied to **objid**s, the values matching the range shall be determined according to clause 7.2.1.3 of the present document, with the exception, that subtrees are not considered.

The inside value matching mechanism *AnyElement*, in addition to types listed in clause B.1.3.1 of ETSI ES 201 873-1 [1], can also be used within objid templates. When applied to objids, it replaces exactly one component.

The inside value matching mechanism *AnyElementsOrNone*, in addition to types listed in clause B.1.3.2 of ETSI ES 201 873-1 [1], can also be used within objid templates. When applied to objids, it matches the longest sequence of components possible, according to the pattern as specified by the components surrounding the "*".

The length restriction matching attribute, in addition to types listed in clause B.1.4.1 of ETSI ES 201 873-1 [1], can also be used with **objid** templates. When applied to objids, it identifies the number of components within an **objid** value matching the **objid** template.

7.2.5 Using **objid** with operators

7.2.5.1 List operator

When ASN.1 is supported, the concatenation operator (&) specified in clause 7.1.2 of ETSI ES 201 873-1 [1] shall be permitted for **objid** values as well. The operation is a simple concatenation of the numerical values of the components from left to right. If necessary (e.g. for logging purposes), the names of the components in the resulted **objid** value shall be determined from the resulted **objid** value (i.e. names of the components will change when the component is changing its position related to the input objid value). The result type is **objid**.

EXAMPLE:

```
 objid{itu_t identified_organization etsi(0)} & objid{inDomain(1) in_Network(1)}
     gives {0 4 0 1 1} that can also be presented as objid{itu_t identified_organization etsi(0)
inDomain(1) in_Network(1)}

 objid{itu_t identified_organization etsi(0)} & objid{iso(1) registration_authority(1)}
     gives {0 4 0 1 1} that can also be presented as objid{itu_t identified_organization etsi(0)
inDomain(1) in_Network(1)}
```

7.2.5.2 Relational operators

It is allowed to use **objid** values as operands of relational operators equality (==), less than (<), greater than (>), non-equality to (!=), greater than or equal to (>=) and less than or equal to (<=).

Two **objid** values are equal, if they have equal number of components and the primary integer values at all positions are the same.

The less than (<), greater than (>), greater than or equal to (>=) and less than or equal to (<=) operations shall use the numerical values of objid value components for the decision, and the decision process shall comply with the following rules:

- the comparison shall start by comparing the first primary integer values of the two objid values and shall be continued in a recursive way until the smaller objid value is found or the two objid values are found to be equal;
- the objid value in which a smaller primary integer value is found first , is less than the other objid value;

- if all compared pairs of primary integer values of the two objid values are equal and one of the objid values has further primary integer values while the other does not, the shorter objid value is less than the longer objid value.

EXAMPLE:

```
// Given
const objid c_etsiMobNet := objid{itu_t identified_organization etsi(0)
                                  mobile_domain(0) umts_Network(1)}
const objid c_etsiINNet  := objid{itu_t identified_organization etsi(0)
                                  inDomain(1) in_Network(1)}
const objid c_etsiIN     := objid{itu_t identified_organization etsi(0)
                                  inDomain(1)}
var objid   v_etsiInIso  := objid{ iso identified_organization dod(6)
                                   internet(1) private(4) enterprise(1) etsi(13019)}
// then
c_etsiMobNet == c_etsiINNet // returns false
c_etsiMobNet <  c_etsiINNet // returns true as the mobile_domain(0) component is numerically
                            // smaller than the inDomain(1) component
c_etsiINNet ==  c_etsiIN    // returns false as c_etsiINNet has more components
c_etsiINNet  >   c_etsiIN   // returns true as c_etsiINNet has more components
v_etsiInIso <= c_etsiMobNet // returns false as the component itu_t(0) is numerically smaller
                            // than the component iso(1))
```

7.2.6 Using **objid** with predefined functions

7.2.6.1 Number of components of an **objid** value or template

In excess the input parameter types given in clause C.28 of ETSI ES 201 873-1 [1], the **lengthof** predefined function shall allow values and templates of **objid** types as input parameter. The actual value to be returned is the sequential number of the last component.

When the function **lengthof** is applied to templates of **objid** types, `inpar` shall only contain the matching mechanisms: SpecificValue, value list, complemented list, *AnyValue*, *AnyValueOrNone*, *AnyElement* and *AnyElementsOrNone* and the length matching attribute. The parameter `inpar` shall only match values, for which the **lengthof** function would give the same result.

Additional error cases are:

- `inpar` can match **objid** values with different number of components.

EXAMPLE:

```
// Given
 var objid v_etsiMobNet := objid{itu_t identified_organization etsi(0)
                                 mobile_domain(0) umts_Network (1)}
 // then
 numElements := lengthof(v_etsiMobNet); // returns 5
```

7.2.6.2 The Substring function

When ASN.1 is supported, the **substr** predefined function shall support **objid** types, i.e. it shall allow objid as type of the input parameter and return an object identifier value containing a fragment (sequence of components) of the input parameter `inpar`. Rules specified in clause C.33 of ETSI ES 201 873-1 [1] shall apply with the following exceptions: `index` zero identifies the first component of the input object identifier value or template. The third input parameter (`count`) defines the number of components in the returned **objid** value.

EXAMPLE:

```
 var objid v_etsiMobNet := objid{itu_t identified_organization etsi(0)
                                 mobile_domain(0) umts_Network (1)}

 substr (v_etsiMobNet, 0, 2)  // returns {itu_t identified_organization}

 substr (v_etsiMobNet, 2, 3)  // returns {etsi(0) mobile_domain(0) umts_Network (1)}
```