ETSI ES 201 873-7 V4.7.1 (2018-05)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

Part 7: Using ASN.1 with TTCN-3

**ETSI Standard**

Reference

RES/MTS-201873-7ed471

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 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.

© ETSI 2018.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.
**3GPP**TM and **LTE**TM are trademarks of ETSI registered for the benefit of its Members and
of the 3GPP Organizational Partners.
**oneM2M** logo is protected for the benefit of its Members.
**GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

# 9 ASN.1 data types and values

## 9.1 Transformation rules for ASN.1 types and values

ASN.1 value sets are handled in the present document the same way as ASN.1 types. Therefore, when referring to "ASN.1 types" in the present document, both ASN.1 value set definitions and type definitions are meant. ASN.1 types and values may be used in TTCN-3 modules. ASN.1 definitions are made using a separate ASN.1 module. ASN.1 types and values are referenced by their type references and value references as produced according to clauses 11.2 and 11.4 of Recommendation ITU-T X.680 [2] within the ASN.1 module(s). Basic ASN.1 value notation and XML ASN.1 value notation shall be transformed equally, i.e. a basic and an XML value notation referring to the same value of the type shall produce the same associated TTCN-3 value.

EXAMPLE 1:

MyASN1module DEFINITIONS ::=

BEGIN

Z::= INTEGER -- Simple type definition

 BMessage::= SEQUENCE -- ASN.1 type definition

 {

 name IA5String,

 title VisibleString,

 date IA5String

 }

 johnValues Bmessage ::= -- ASN.1 value definition

 {

 name "John Doe",

 title "Mr",

 date "April 12th"

 }

 johnValuesXML ::= -- XML ASN.1 value definition

 <Bmessage>

 <name>John Doe</name>

 <title>Mr</title>

 <date>April 12th<date>

 <Bmessage>

 DefinedValuesForField1 Z ::= {0 | 1} -- ASN.1 subtype definition

END

The ASN.1 module shall conform to the syntax and semantics of the Recommendations ITU-T X.680 [2], X.681 [3], X.682 [4] and X.683 [5]. Once declared and imported, ASN.1 types and values may be used within TTCN-3 modules in a similar way than TTCN-3 types and values, imported from other TTCN-3 modules. Each imported ASN.1 definition produces an associated type or value. All TTCN-3 definitions or assignments based on imported ASN.1 definitions shall be done according the rules imposed by the related associated type or value. Also, the matching mechanism shall use the associated type when matching at a receiving or a **match** operation.

Associated types and values are derived from ASN.1 definitions by applying the transformation rules below. Transformations shall be started on a valid ASN.1 module and end in a valid TTCN-3 representation. The order corresponds to the order of execution of the individual transformations:

0) Ignore all type prefixes and all encoding control sections (see note 1). ASN.1 type prefixes may consist of tags and encoding prefixes.

0bis) Ignore names of inner types and values in SEQUENCE OF and SET OF definitions (see note 1).

1/a) Ignore any extension markers and exception specifications.

1/b) All fields in extension additions of a SEQUENCE type or a SET type shall be transformed to optional fields.

NOTE: Tool vendors are encouraged to check the additional type restrictions imposed by the extension for TTCN-3 values and templates.

2) Ignore any user defined constraints (see clause 9 of Recommendation ITU-T X.682 [4]).

3) Ignore any contents constraint (see clause 11 of Recommendation ITU-T X.682 [4]).

4) Convert pattern constraints (see clause 51.9 of Recommendation ITU-T X.680 [2]) to TTCN-3 pattern subtypes (see clause 6.1.2.5 of ETSI ES 201 873-1 [1]).

5) Execute the COMPONENTS OF transformation according to clause 25.5 of Recommendation ITU-T X.680 [2] on any SEQUENCE types and according to clause 27.2 on any SET types containing the keywords "COMPONENTS OF".

6) Create equivalent TTCN-3 subtypes for all ASN.1 types constrained using contained subtyping by replacing included types by the set of values they represent. More detailed information on the conversion of ASN.1 type constraints to TTCN-3 subtypes is given in table 4. Table 4 shows the applicability of ASN.1 type constraint mechanisms to different ASN.1 types. Where the cell contains "No", the type constraint is disallowed for the given type. Shaded cells identify type constraints applicable to a given type and text in the cell defines TTCN‑3 subtyping mechanisms to be used when transforming constrained ASN.1 types.

7) Replace any EMBEDDED PDV type with its associated type obtained by expanding inner subtyping in the associated type of the EMBEDDED PDV type (see clause 36.5 of Recommendation ITU-T X.680 [2]) to a full type definition.

8) Replace the EXTERNAL type with its associated type obtained by expanding inner subtyping in the associated type of the EXTERNAL type (see clause 37.5 of Recommendation ITU-T X.680 [2]) to a full type definition (see note 3).

9) Replace the CHARACTER STRING type with its associated type obtained by expanding inner subtyping in the associated type of the CHARACTER STRING type (see clause 44.5 of Recommendation ITU-T X.680 [2]) to a full type definition.

10) Replace the INSTANCE OF type with its associated type obtained by substituting INSTANCE OF DefinedObjectClass by its associated ASN.1 type (see clause C.7 of Recommendation ITU-T X.681 [3]) and replace all ASN.1 types with their TTCN-3 equivalents according to table 3. The resulted type is the TTCN-3 associated type.

11) Ignore any remaining inner subtyping (see note 4).

12) For each named number and each named bit a TTCN-3 constant definition shall be generated: the type of the constant is the TTCN-3 type corresponding to the ASN.1 type being translated and the name of the constant is constructed as: the name of the TTCN-3 type corresponding to the given ASN.1 type, followed by a "**\_**" (LOW LINE) character, followed by the name of the number or named bit converted according to clause 8.2 of the present document, followed by a "**\_**" (LOW LINE) character. Named number constant values will be the TTCN-3 equivalent of the ASN.1 number that is named. Named bit constants for named bits introduced by the same type shall all have the same length (determined as the maximum of the minimal type length and the maximal named bit number). In ASN.1 values replace any named number by its value and substitute any named bits or sequence of named bits by a bitstring with appended "0 "s at the end for minimal length of the BIT STRING type, where bit positions identified by names present are replaced by "1"s, other bit positions are replaced by "0"s.

EXAMPLE 2:

 -- The definition in ASN.1:

 Color ::= INTEGER {red(0),green(1), blue(255) }

 // is mapped to the TTCN-3 type and constants:
 **type** integer Color;

 **const** Color Color\_red\_ := 0;

 **const** Color Color\_green\_ := 1;

 **const** Color Color\_blue\_ := 255;

EXAMPLE 3:

 -- The definition in ASN.1:

 Workdays ::=

 BIT STRING {monday(0),tuesday(1), wednesday(3), thursday(4), friday(5) } (SIZE(7))

 // is mapped to the TTCN-3 type and constants:
 **type** bitstring Workdays;

 **const** Workdays Workdays\_monday\_ := '1000000'B;

 **const** Workdays Workdays\_tuesday\_ := '0100000'B;

 **const** Workdays Workdays\_wednesday\_ := '0001000'B;

 **const** Workdays Workdays\_thursday\_ := '0000100'B;

 **const** Workdays Workdays\_friday\_ := '0000010'B;

 // the following ASN.1 bitstring value notation

 {Monday, Wednesday, Friday}

 // is mapped to the TTCN-3 value:

 '1001010'B

13) Replace any selection type with the type referenced by the selection type; if the denoted choice type (the "Type" in clause 30.1 of Recommendation ITU-T X.680 [2]) is a constrained type, the selection has to be done on the parent type of the denoted choice type.

14) Convert any RELATIVE-OID type or value to an **objid** type or value (see note 5).

15) Replace any of the following restricted character string types with their associated types obtained as (see note 6):

* BMPString: **universal charstring** (char ( 0,0,0,0 ) .. char ( 0,0,255,255));
* UTF8String: **universal charstring**;
* NumericString: **charstring** constrained to the set of characters as given in clause 41.2 of Recommendation ITU-T X.680 [2];
* PrintableString: **charstring** constrained to the set of characters as given in clause 41.4 of Recommendation ITU-T X.680 [2];
* TeletexString and T61String: **universal** **charstring** constrained to the set of characters as given in clause 41.1, table 8, row TeletexString (T61String) of Recommendation ITU‑T X.680 [2],inclause 9.1 of the present document;
* VideotexString: **universal charstring** constrained to the set of characters as given in Recommendations ITU-T T.100 [9] and T.101 [10];
* GraphicString: **universal** **charstring**;
* GeneralString: **universal** **charstring**.

16) Replace any of the following time types with their associated types obtained as (see notes 14 and 15):

* GeneralizedTime types or values with the type or value of **charstring**;
* UTCTime types or values with the type or value of **charstring**;
* TIME, DATE, TIME-OF-DAY, DATE-TIME and DURATION types or values with the type or value of **charstring**. Properties settings, inner and range subtyping, if any, shall be ignored (see also table 4).
1. Replace any of the following types with their associated types obtained as:
* ObjectDescriptor type or value by the **universal** **charstring** type or value;
* OID-IRI type or value by the **universal** **charstring** type or value (see note 14);
* RELATIVE-OID-IRI type or value by the **universal** **charstring** type or value (see note 14).

18) Replace any notations for the object class field types (see clause 14 of Recommendation ITU-T X.681 [3]) by the ASN.1 definition they are referring to (see note 8); open types has to be replaced by the metatype "OPEN TYPE" for the purpose of the transformation (and only for that).

19) Replace all information from objects notations (see clause 15 of Recommendation ITU-T X.681 [3]) by the ASN.1 definition they are referencing to.

20) Revert table constraints (see clause 10 of Recommendation ITU-T X.682 [4]) to list subtyping and ignore all relational constraints (see note 7).

21) Replace all occurrences of NULL type with the following associated TTCN-3 type (see note 13):

* **type enumerated** *<identifier>* { NULL },
where *<identifier>* is the ASN.1 Type reference converted according to clause8.2, if a synonym of the NULL type is defined; or with
* the nested type definition **enumerated { NULL }** *<identifier>,*where *<identifier>* is the ASN.1 field identifier, converted according to clause 8.2, if the ASN.1 **NULL** type is used within a structured type.

22) Replace all references to open types with the metatype "OPEN TYPE" (see note 11).

23) Replace ASN.1 types with their equivalents according to table 3 and ASN.1 values with equivalent TTCN‑3 values based on the associated types. Fields of ASN.1 SEQUENCE and SET types identified as OPTIONAL or with a DEFAULT value shall be optional fields in the associated type (see note 12). Missing (i.e. implicitly omitted) optional fields in structured ASN.1 values (of the types (SET, SEQUENCE, etc.) shall be explicitly omitted in the resulted structured TTCN-3 values (see note 9).

24) Replace the metatype "OPEN TYPE" by **anytype**.

NOTE 1: Associated types and values contain abstract information only, thus do not contain all information needed for correct encoding. The way of handling the information needed by the test system to provide correct encoding and/or decoding (both embedded in ASN.1 definitions and provided in the encoding reference default, tag default and extension default settings of ASN.1 modules and encoding control sections, if any and information coming from the ASN.1 specification itself, like tag values of built-in ASN.1 types) is implementation dependent and remains hidden for the user; this knowledge is not required to make valid TTCN‑3 declarations or assignments involving imported ASN.1 types and values.

NOTE 2: When importing ENUMERATED types, integer numbers assigned by the user to enumerations will also be imported.

NOTE 3: The data-value field of the EXTERNAL type may be encoded as a single-ASN1-type, octet-aligned or arbitrary (see clause 8.18.1 of Recommendation ITU-T X.690 [6]) at the discretion of the encoder; if the user wants to enforce one given form of encoding or wants to allow only one specific encoding form at matching, it has to use the appropriate encoding attribute for the type or the given constant, variable, template or template field (see clause 11.3 of the present document).

NOTE 4: Inner subtyping has to be taken into account by the user when defining TTCN-3 values or templates based on an ASN.1 type constrained by inner subtyping.

NOTE 5: Equivalence with the **objid** type is limited to the syntax to be used for value notations only. When encoding/decoding an **objid** value retrieved from an ASN.1 RELATIVE-OID value using an ASN.1 encoding rule, the encoding/decoding will occur according to rules specified for the RELATIVE-OID type.

NOTE 6: VisibleString, IA5String and UniversalString have their equivalent TTCN-3 types and are replaced directly.

NOTE 7: Relational constraints have to be taken into account by the user when declaring values and templates (also may be handled by tools implicitly).

NOTE 8: This replacement does not affect constraints applied to the "notation for the object class field type" itself.

NOTE 9: Missing optional fields in values of structured ASN.1 types (SET, SEQUENCE, EXTERNAL, etc.) are equivalent to explicitly omitted fields in structured TTCN-3 values.

EXAMPLE 4:

 module MyTTCNModule

 {

 import from MyASN1module language "ASN.1:2002" all;

 const Bmessage MyTTCNConst:= johnValues;

 const DefinedValuesForField1 Value1:= 1;

 }

NOTE 10: ASN.1 definitions other than types and values (i.e. information object classes or information object sets) are not directly accessible from the TTCN-3 notation. Such definitions will be resolved to a type or value within the ASN.1 module before they can be referenced from within the TTCN-3 module.

NOTE 11: The metatype "OPEN TYPE" is just used to describe the transformation process. It does not exist neither before nor after the transformation.

NOTE 12: Most ASN.1 encoding rules require that fields with DEFAULT values are omitted in the encoded message when their actual contents equal to the default values. However, in TTCN-3, it may be required that the default value is also encoded and present. If fields with default values are omitted or present in the encoded message, is a TTCN-3 test system runtime configuration option. It is also a TTCN-3 test system runtime configuration option, if fields with default values missing in the received encoded message are omitted or substituted by their default values in the abstract TTCN-3 value (the decoded message).

NOTE 13: The associated type for the ASN.1 NULL type is introduced to specify the TTCN-3 value notation for this type. The encoding/decoding of NULL values and fields have to be as defined for the NULL type in the ASN.1 Recommendations (see e.g. in Recommendations ITU-T X.690 [6], X.691 [7] and X.693 [8]). Also, the restriction in clause 7.1.3 of ETSI ES 201 873-1 [1] (relational operators) that only values of the same enumerated types are allowed to be compared, does not apply to imported ASN.1 NULL types.

NOTE 14: The ASN.1 time types (including its useful types) are transformed to a restricted TTCN-3 **charstring** type and the OID-IRI and RELATIVE-OID-IRI types are transformed to the TTCN‑3 **universal charstring** type primarily for tool efficiency reasons (though this approach also allows sending some invalid values without the need to create a specific type). This, however, means that the user should exercise specific caution in receiving templates, as AnyValue and AnyValuesOrNone will accept incorrectly formatted time values as well. When the correctness of the received values is important, the **pattern** matching (possibly appended with the **ifpresent** matching attribute for optional fields) should be used instead of AnyValue and AnyValuesOrNone. TTCN-3 patterns for the time types are given in annex E and for the OID-IRI type is given in clause C.2.

NOTE 15: Though all ASN.1 time types are transformed to a restricted TTCN-3 **charstring** type, they differ in their tag values (see clauses and 38.1.1, 38.4, 46.3 and 47.3 of Recommendation ITU-T X.680 [2]) and encodings (see clauses 8.25 and 8.26 of Recommendation ITU-T X.690 [6] and clauses 10.6.5 and 32 of Recommendation ITU-T X.691 [7]). Therefore it is necessary that TTCN-3 tools retain the type information and encode the values accordingly.

Table 4: ASN.1 type constraint to TTCN-3 subtype conversions

| Type (or derived from such a type by tagging or subtyping) | Single Value | Contained Subtype (h) | Value Range | Size Constraint | Permitted Alphabet | Type Constraint | Inner Subtyping(see i) | Pattern Constraint | User defined constraint | Table constraint(see k) | Relation constraint(see k) | Content constraint | Propertysettings |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit String | list | single value: list, size: length | No | length | No | No | No | No | ignore | No | No | ignore | No |
| Boolean | list | list | No | No | No | No | No | No | ignore | No | No | No | No |
| Choice | list | list | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Embedded-pdv (see note a) | list | No | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Enumerated | list | list | No | No | No | No | No | No | ignore | No | No | No | No |
| External (see a) | list | No | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Instance-of (see a and b) | list | list | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Integer | list | single value: list, value range: range | range and/or list (l,m) | No | No | No | No | No | ignore | No | No | No | No |
| Null | ignore | ignore | No | No | No | No | No | No | ignore | No | No | No | No |
| Object class field type | (see c) | (see c) | No | No | No | No | No | No | ignore | list | ignore | No | No |
| Object Descriptor (see e) | list | single value: list, size: length, perm.alphabet: range | No | length | range | No | No | No | ignore | No | No | No | No |
| Object Identifier | list | list | No | No | No | No | No | No | ignore | No | No | No | No |
| Octet String | list | single value: list, size: length | No | length | No | No | No | No | ignore | No | No | ignore | No |
| Open type | No | No | No | No | No | anytype with list constraint | No | No | ignore | No(see m) | No(see m) | No | No |
| Real | list | single value: list, value range: range | range and/or list (n,o) | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Relative Object Identifier(see d) | list | list | No | No | No | No | No | No | ignore | No | No | No | No |
| Restricted Character String Types | list | single value: list, size: length, perm.alpha­bet: range | range | length | range | No | No | Ignore(see g) | ignore | No | No | No | No |
| Sequence | list | list | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Sequence-of | list | single value: list, value range: range | No | length | No | No | convert to full type | No | ignore | No | No | No | No |
| Set | list | list | No | No | No | No | convert to full type | No | ignore | No | No | No | No |
| Set-of | list | single value: list, value range: range | No | length | No | No | convert to full type | No | ignore | No | No | No | No |
| TIME <including its derivations> | list | list | ignore(p) | No | No | No | ignore | No | No | No | No | No | ignore |
| Time Types (see a) | list | list | No | No | No | No | No | No | ignore | No | No | No | No |
| Unrestricted Character String Type (see a) | list | No | No | length (applied to field "string-value") | No | No | convert to full type | No | ignore | No | No | No | No |
| NOTES:(a) These types are seen from TTCN-3 as being equivalent to their associated types.(b) Type-id field of the associated type for Instance of shall be replaced by the type of the &id field the value field is anytype (annex C of Recommendation ITU-T X.681 [3]).(c) Replaced by the referenced type, thus applicable as to the referenced type.(d) Seen as object identifier from TTCN-3.(e) Its associated type is a restricted character string type.(f) Open type is replaced by **anytype**.(g) Character patterns can only be used in constants, variables, templates and module parameters in TTCN-3 but cannot be used for subtyping.(h) Contained subtype constraints shall be replaced by literal constraints at import.(i) Information in this column relates to the TTCN-3 views of ASN.1 definitions. Encoding/decoding shall be according to the root type, thus extra information for encoding also has to be stored which are not shown in this table.(j) Applicable to notations for the object class field type only.(k) Applicable when the open type is defined using the notation for the object class field type (see j)).(l) If the lower and the upper boundaries of an ASN.1 range equal, the range shall be translated to a TTCN‑3 list subtyping, corresponding to the allowed ASN.1 value (please note, there may be more than one ranges in an ASN.1 range subtype specification). |
| (m) For each range, not obeying rule (l) above a TTCN‑3 subtype range shall be generated, considering the following:If the lower boundary of an ASN.1 range is MIN, the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the lower boundary of the ASN.1 type's parent type or **‑infinity** (if the parent type is a built-in type or has no lower boundary, i.e. MIN - either open or closed - is used along the whole derivation chain).If the lower boundary of an ASN.1 range is MIN<, the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the lower boundary of the ASN.1 type's parent type plus 1 or **‑infinity** (if the parent type is a built-in type or has no lower boundary, i.e. MIN - either open or closed - is used along the whole derivation chain).If the lower boundary of an ASN.1 range is a value and is a closed boundary (i.e. not MIN and does not include the "<" symbol), the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the ASN.1 lower boundary.If the lower boundary of an ASN.1 range is a value and is an open boundary (i.e. not MIN< and does include the "<" symbol), the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the ASN.1 lower boundary plus 1. If the upper boundary of an ASN.1 range is MAX, the upper boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the upper boundary of the ASN.1 type's parent type or **infinity** (if the parent type is a built-in type or has no upper boundary, i.e. MAX - either open or closed - are used along the whole derivation chain).If the upper boundary of an ASN.1 range is <MAX, the upper boundary of the TTCN‑3 range shall be inclusive and its value shall be the upper boundary of the ASN.1 type's parent type minus 1 or **infinity** (if the parent type is a built-in type or has no upper boundary, i.e. MAX - either open or closed - are used along the whole derivation chain).If the upper boundary of an ASN.1 range is a value and is a closed boundary (i.e. not MAX and does not include the "<" symbol), the upper boundary of the TTCN‑3 range shall be inclusive and its value shall be the ASN.1 upper boundary.If the upper boundary of an ASN.1 range is a value and is an open boundary (i.e. not <MAX and does include the "<" symbol), the upper boundary of the TTCN‑3 range shall be inclusive and its value shall be the ASN.1 upper boundary minus 1.(n) If the lower and the upper boundaries of an ASN.1 range equal, the range shall be translated to a TTCN‑3 list subtyping, corresponding to the allowed ASN.1 value (please note, there may be more than one ranges in an ASN.1 range subtype specification).(o) For each range, not obeying rule (n) above a TTCN‑3 subtype range shall be generated, considering the following:If the lower boundary of an ASN.1 range is MINUS-INFINITY, the lower bound of the corresponding TTCN‑3 range shall be **-infinity**.If the lower boundary of an ASN.1 range is MIN, the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the lower boundary of the ASN.1 type's parent type or **‑infinity** (if the parent type is a built-in type or has no lower boundary, i.e. MIN and/or MINUS-INFINITY - either open or closed - are used along the whole derivation chain).If the lower boundary of an ASN.1 range is MIN<, the lower boundary of the corresponding TTCN‑3 range shall be either an exclusive value equivalent to the lower boundary of the ASN.1 type's parent type or **‑infinity** (if the parent type is a built-in type or has no lower boundary, i.e. MIN and/or MINUS-INFINITY - either open or closed - are used along the whole derivation chain).If the lower boundary of an ASN.1 range is a numerical value and is a closed boundary (i.e. not MIN or MINUS-INFINITY and does not include the "<" symbol), the lower boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the ASN.1 lower boundary.If the lower boundary of an ASN.1 range is a numerical value and is an open boundary (i.e. not MIN< or MINUS-INFINITY< and does include the "<" symbol), the lower boundary of the corresponding TTCN‑3 range shall be exclusive and its value shall be the ASN.1 lower boundary. If an upper boundary of an ASN.1 range is NOT-A-NUMBER, a TTCN‑3 range with the upper boundary **infinity** and the list subtype value **not\_a\_number** shall be generated for this range.If an upper boundary of an ASN.1 range is <NOT-A-NUMBER, a TTCN‑3 range with the upper boundary **infinity** shall be generated for this range.If the upper boundary of an ASN.1 range is PLUS-INFINITY, the upper bound of the corresponding TTCN‑3 range shall be **infinity**.If the upper boundary of an ASN.1 range is MAX, the upper boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the upper boundary of the ASN.1 type's parent type or **infinity** (if the parent type is a built-in type or has no upper boundary, i.e. MAX, PLUS-INFINITY orNOT-A-NUMBER - either open or closed - are used along the whole derivation chain).If the upper boundary of an ASN.1 range is <MAX, the upper boundary of the corresponding TTCN‑3 range shall be either an exclusive value equivalent to the upper boundary of the ASN.1 type's parent type or **infinity** (if the parent type is a built-in type or has no upper boundary, i.e. MAX, PLUS-INFINITY orNOT-A-NUMBER - either open or closed - are used along the whole derivation chain).If the upper boundary of an ASN.1 range is a numerical value and is a closed boundary (i.e. not MAX, PLUS-INFINITY or NOT-A-NUMBER and does not include the "<" symbol), the upper boundary of the corresponding TTCN‑3 range shall be inclusive and its value shall be the ASN.1 upper boundary.If the upper boundary of an ASN.1 range is a numerical value and is an open boundary (i.e. not <MAX, <PLUS-INFINITY or <NOT-A-NUMBER and does include the "<" symbol), the upper boundary of the corresponding TTCN‑3 range shall be exclusive and its value shall be the ASN.1 upper boundary.(p) In case of TIME types range subtyping means duration, time point and recurrance ranges rather than value range. |

ASN.1 allows using set arithmethics in subtype constraints. In many cases these set arithmetics expressions shall be calculated before the translation to be able to convert the ASN.1 subtype into its associated TTCN-3 type. However, in some cases expressions can be translated to TTCN-3 in different ways; e.g. a UNION of consequtive integer values can be translated into a list of values, into a value range or a mixture of the two, while all these forms denote the same set of values. The present document leaves this choice open for tool implementations. The only requirement is that the resulted associated TTCN-3 type shall conform to this clause and clause 6 of of ETSI ES 201 873‑1 [1].

## 9.2 Transformation rules for values

In case of real values, the base used in the value notation (2 or 10) shall be retained by the tool to be able to produce the correct encoding of the value. However, from the point of view of TTCN-3 relational operations only the numerical value counts.

## 9.3 Scope of ASN.1 identifiers

Imported ASN.1 identifiers follow the same scope rules as imported TTCN-3 types and values (see clause 5.2 of ETSI ES 201 873‑1 [1]).