



Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 6: Mapping to TTCN-3

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard: https://standards.iteh.ai/catalog/standards/sis/6531660-6d27-4e1e-bde9-2fb0dea28017/etsi-es-203-119-6-v1-2-20-08

ReferenceRES/MTS-TDL1-6v121

Keywordsmethodology, model, testing, TTCN-3

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

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 2020.

All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

GSM® and the GSM logo are trademarks 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.....	8
3 Definition of terms, symbols and abbreviations.....	9
3.1 Terms.....	9
3.2 Symbols.....	9
3.3 Abbreviations	9
4 Basic Principles	9
4.1 Introduction	9
4.2 Document Structure.....	9
4.3 Notational Conventions	10
4.3.0 General.....	10
4.3.1 Functions used in production rules	11
4.3.2 Predefined Annotations.....	11
4.4 Conformance	11
5 Foundation.....	11
5.1 Overview	11
5.2 Mapping of Foundation Elements	11
5.2.1 Element	11
5.2.2 NamedElement	12
5.2.3 PackageableElement	12
5.2.4 Package	12
5.2.5 ElementImport	12
5.2.6 Comment	13
5.2.7 Annotation	13
5.2.8 AnnotationType	13
5.2.9 TestObjective.....	13
5.2.10 Extension	13
6 Data	14
6.1 Overview	14
6.2 Mapping of Data Definition Elements.....	14
6.2.1 DataResourceMapping.....	14
6.2.2 MappableDataElement.....	14
6.2.3 DataElementMapping	14
6.2.4 ParameterMapping.....	14
6.2.5 DataType	14
6.2.6 DataInstance	14
6.2.7 SimpleDataType	14
6.2.8 SimpleDataInstance	15
6.2.9 StructuredDataType	15
6.2.10 Member.....	15
6.2.11 StructuredDataInstance	15
6.2.12 MemberAssignment.....	16
6.2.13 CollectionDataType	16
6.2.14 CollectionDataInstance	16
6.2.15 ProcedureSignature	16
6.2.16 ProcedureParameter	16
6.2.17 ParameterKind	16
6.2.18 Parameter	16

6.2.19	FormalParameter.....	17
6.2.20	Variable	17
6.2.21	Action	17
6.2.22	Function	17
6.2.23	UnassignedMemberTreatment.....	18
6.2.24	PredefinedFunction.....	18
6.3	Mapping of Data Use Elements.....	18
6.3.1	DataUse	18
6.3.2	ParameterBinding	18
6.3.3	StaticDataUse	18
6.3.4	DataInstanceUse	18
6.3.5	SpecialValueUse.....	21
6.3.6	AnyValue.....	21
6.3.7	AnyValueOrOmit	21
6.3.8	OmitValue.....	21
6.3.9	DynamicDataUse	21
6.3.10	FunctionCall	21
6.3.11	FormalParameterUse	22
6.3.12	VariableUse	24
6.3.13	PredefinedFunctionCall	25
6.3.14	LiteralValueUse	25
7	Time	26
7.1	Overview	26
7.2	Mapping of Time Elements	26
7.2.1	Time.....	26
7.2.2	TimeLabel.....	26
7.2.3	TimeLabelUse.....	27
7.2.4	TimeLabelUseKind.....	27
7.2.5	TimeConstraint	27
7.2.6	TimeOperation.....	28
7.2.7	Wait	29
7.2.8	Quiescence.....	29
7.2.9	Timer	29
7.2.10	TimerOperation.....	29
7.2.11	TimerStart.....	29
7.2.12	TimerStop	30
7.2.13	TimeOut.....	30
8	Test Configuration.....	30
8.1	Overview	30
8.2	Mapping of TestConfiguration Elements in Non-special Cases.....	30
8.2.1	Introduction.....	30
8.2.2	GateType	31
8.2.3	GateTypeKind.....	31
8.2.4	GateInstance	31
8.2.5	ComponentType	31
8.2.6	ComponentInstance	32
8.2.7	ComponentInstanceRole	32
8.2.8	GateReference.....	32
8.2.9	Connection	32
8.2.10	TestConfiguration	33
8.2.11	Definition of the component type of MTC	33
8.2.12	Definition of the component type of system.....	33
8.3	Mapping of TestConfiguration Elements in Special Cases	34
8.3.1	Introduction.....	34
8.3.2	Connectable and mappable GateType.....	34
8.3.3	A gate connected to a Tester and an SUT.....	34
8.3.4	More than one SUT.....	35
8.3.5	A gate of a Tester is connected to more SUTs.....	36
8.3.6	A gate is connected to more gates of the same component.....	36
9	Test Behaviour	37

9.1	Overview	37
9.2	Mapping of Test Description Elements	38
9.2.1	TestDescription	38
9.2.2	BehaviourDescription	40
9.3	Mapping of Combined Behaviour elements	40
9.3.1	Behaviour	40
9.3.2	Block	40
9.3.3	LocalExpression	40
9.3.4	CombinedBehaviour	40
9.3.5	SingleCombinedBehaviour	40
9.3.6	CompoundBehaviour	40
9.3.7	BoundedLoopBehaviour	41
9.3.8	UnboundedLoopBehaviour	41
9.3.9	OptionalBehaviour	41
9.3.10	MultipleCombinedBehaviour	43
9.3.11	AlternativeBehaviour	44
9.3.12	ConditionalBehaviour	44
9.3.13	ParallelBehaviour	44
9.3.14	ExceptionalBehaviour	45
9.3.15	DefaultBehaviour	47
9.3.16	InterruptBehaviour	47
9.3.17	PeriodicBehaviour	47
9.4	Mapping of Atomic Behaviour Elements	47
9.4.1	AtomicBehaviour	47
9.4.2	Break	47
9.4.3	Stop	48
9.4.4	VerdictAssignment	48
9.4.5	Assertion	48
9.4.6	Interaction	48
9.4.7	Message	48
9.4.8	ProcedureCall	49
9.4.9	Target	51
9.4.10	ValueAssignment	51
9.4.11	TestDescriptionReference	51
9.4.12	ComponentInstanceBinding	52
9.4.13	ActionBehaviour	52
9.4.14	ActionReference	52
9.4.15	InlineAction	52
9.4.16	Assignment	52
10	Predefined TDL Model Instances	53
10.1	Overview	53
10.2	Mapping of Predefined Instances of the 'SimpleDataType' Element	53
10.2.1	Boolean	53
10.2.2	Integer	53
10.2.3	String	53
10.2.4	Verdict	53
10.3	Mapping of Predefined Instances of 'SimpleDataInstance' Element	53
10.3.1	true	53
10.3.2	false	53
10.3.3	pass	53
10.3.4	fail	53
10.3.5	inconclusive	53
10.4	Mapping of Predefined Instances of 'Time' Element	54
10.4.1	Second	54
10.5	Mapping of Predefined Instances of the 'Function' Element	54
10.5.1	Overview	54
10.5.2	Functions of Return Type 'Boolean'	54
10.5.3	Functions of Return Type 'Integer'	55
10.5.4	Functions of Return Type of Instance of 'Time'	56
Annex A (informative):		
	 Examples of mapping TDL to TTCN-3	57

A.1	Introduction	57
A.2	A 3GPP Conformance Example in Textual Syntax	57
A.3	An IMS Interoperability Example in Textual Syntax.....	62
	History	69

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/65950a60-6d27-4e1e-bde9-2fb0dea28017/etsi-es-203-119-6-v1.2.1-2020-08>

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

The present document is part 6 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 specifies how the elements of the Test Description Language (TDL) should be mapped to Testing and Test Control Notation version 3 (TTCN-3) [2]. The intended use of the present document is to serve as the basis for the development of TDL tools. The meta-model of TDL and the meanings of the meta-classes are described in ETSI ES 203 119-1 [1].

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 203 119-1 (V1.5.1): "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 1: Abstract Syntax and Associated Semantics".
- [2] ETSI ES 201 873-1 (V4.9.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".
- [3] ETSI ES 203 119-3 (V1.4.1): "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 3: Exchange Format".

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] ETSI TS 136 523-1 (V10.2.0): "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Packet Core (EPC); User Equipment (UE) conformance specification; Part 1: Protocol conformance specification (3GPP TS 36.523-1 version 10.2.0 Release 10)".
- [i.2] ETSI TS 186 011-2: "Core Network and Interoperability Testing (INT); IMS NNI Interoperability Test Specifications (3GPP Release 10); Part 2: Test descriptions for IMS NNI Interoperability".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

behaviour function: function used in TTCN-3 code that describes the behaviour of a TDL component instance

TTCNname: name of a TDL meta-model element that is used in the TTCN-3 code

NOTE: A TTCNname of a TDL element follows the syntactical rules of identifiers specified in ETSI ES 201 873-1 [2]. A TTCNname of a TDL element may contain a part that is derived from the TDL name with some prefixes and/or postfixes determined by a naming convention used in the TTCN-3 code.

3.2 Symbols

Void.

3.3 Abbreviations

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

IMS	IP Multimedia Subsystem
MTC	Main Test Component
PTC	Parallel Test Component
SUT	System Under Test
TDL	Test Description Language
TTCN-3	Testing and Test Control Notation version 3

4 Basic Principles

4.1 Introduction

While both TDL and TTCN-3 are standardized languages, there are various ways how TTCN-3 code can be derived from a TDL test description. This may result in different or even incompatible code intended to implement the same test description. Without a standardized mapping of TDL to TTCN-3, there could be a proliferation of different and possibly incompatible tool- and user-specific mappings of TDL test descriptions to executable test cases which can present new challenges to users and tool vendors.

A standardized mapping between the two languages provides a consistent approach for producing executable tests from high level test descriptions specified in TDL. This enables the generation of executable tests from TDL test descriptions in a (semi-) automatic way, and by extension of the re-use of existing test tools and frameworks for test execution. This way, test engineers can concentrate on the specification of test descriptions at a higher level of abstraction, while having a clear expectation of what the resulting test implementation will look like.

4.2 Document Structure

The meta-model of the Test Description Language is specified in ETSI ES 203 119-1 [1]. The present document specifies how the elements of the meta-model of TDL in locally ordered 'TestDescriptions' should be mapped to TTCN-3 code. The mapping of the globally ordered 'TestDescription's is outside the scope of the present document.

The structure of the present document follows the structure of the meta-model specification in ETSI ES 203 119-1 [1]. The clauses 5 to 10 describe the standardized mappings of the meta-model elements with identical clause numbers in ETSI ES 203 119-1 [1]. In each clause, first the description of the mapping of the corresponding meta-model element is described. It may be followed by a Constraints section, if the mapping is provided only with limitations. At the end of a clause an Example clause may exist to illustrate the mapping of the corresponding meta-model element. In the Examples the textual (specified in ETSI ES 203 119-1 [1]) or the graphical (specified in ETSI ES 203 119-3 [3]) notations of TDL can be used.

In some cases the structure of the TTCN-3 code may differ from the structure of the TDL specification or it requires some additional specification in TTCN-3. These special cases are described in clauses 8.2.11, 8.2.12 and 8.3.

At the end of the present document in Annex A several examples illustrate how the TTCN-3 code will look like after the rules of mapping specified in the present document are applied.

4.3 Notational Conventions

4.3.0 General

Elements (e.g. meta-classes, properties, etc.) from the TDL meta-model [1] are typed in between 'single quotes', e.g. 'StructuredDataType' or 'returnType'.

The TTCN-3 code elements (keywords, symbols, etc.) are typed in **bold Courier New** font, e.g. **type port** or **{**.

The TTCN-3 code to be generated is described by production rules, where applicable. The production rules are specified in between << and >> symbols. Inside a production rule, the concatenation between elements of that production rule is specified by a plus (+) symbol.

Iterations over collections of attributes of a metaclass make use of a function collect() with the following syntax: *propertyName.collect(VariableName ':' expression)*, where *VariableName* is an alphanumeric word signifying the variable used in the subsequent *expression*, *propertyName* is a string that shall be the same as the name of a property of a TDL metaclass. The type of this property determines the type of the variable denoted by *VariableName*.

The separator between the elements of an iteration is specified by the concat() function.

EXAMPLE 1:

The production rule:

```
type record <<self.name>> {
    << member.collect(m | m.dataType.name + " " + m.name() ).concat(",")>>
}
```

for this TDL description

Type MSG (sessionID of type integer, content of type charstring);

will provide the following TTCN-3 code snippet:

```
type record MSG {
    integer sessionID,
    charstring content
}
```

The function select() selects a TDL element with a given value of a property.

EXAMPLE 2:

`componentInstance.select(c | c.role = Tester)` selects a 'componentInstance' whose 'role' property has a value of 'Tester'.

Other helper functions used in the production rules are collected in clause 4.3.1, while the predefined 'AnnotationType's that can be used to control the TTCN-3 code generation are listed in clause 4.3.2.

4.3.1 Functions used in production rules

- `behaviourFunctionInReferencedTD()`: returns the name of the behaviour function used in a referenced 'TestDescription' of the same tester component.
- `equivalent()`: returns the equivalent of the corresponding TDL element. If none of the structural modifications - described in clause 8.3 - on a TDL configuration is to be applied then the `element.equivalent()` is the element itself, otherwise what is specified in the corresponding sub-clause of clause 8.3.
- `getKind()`: returns the kind of an 'importedElement' (e.g. **type**, **template**, **const**, **function**, etc.) that can be used in a TTCN-3 **import** statement.
- `toLower()`: returns the value of a literal converted to all lowercase characters.
- `TTCNname()`: returns the name of the corresponding TDL element that will be used in the TTCN-3 code.

4.3.2 Predefined Annotations

A Predefined Annotation is an 'Annotation', whose 'key' is one of the following predefined 'AnnotationTypes'. The Predefined Annotations are used to help the TTCN-3 code generation in cases where the TTCN-3 code to be generated cannot be determined just from the TDL description:

- `TTCN3Code`: this 'AnnotationType' indicates that the 'body' of the 'Annotation' or of an 'InlineAction' contains a valid TTCN-3 code.
- `Value`: this 'AnnotationType' indicates that the annotated element shall not be treated as a template or a template type.

4.4 Conformance

For an implementation claiming to conform to this version of the mapping from TDL to TTCN-3, all features specified in the present document and in ETSI ES 203 119-1 [1] shall be implemented consistently with the requirements given in the present document and ETSI ES 203 119-1 [1].

5 Foundation

5.1 Overview

'Package's are mapped to TTCN-3 modules, 'ElementImport's to import statements, while 'Comment's, 'Annotation's, 'AnnotationType's and 'TestObjective's to TTCN-3 comments.

5.2 Mapping of Foundation Elements

5.2.1 Element

This is an abstract metaclass, therefore no mapping is defined.

Naming is different in TDL and in TTCN-3, therefore the names of the 'Element's used in TDL may not be used in TTCN-3. On one hand the set of characters allowed to be used in a TDL name is larger than the set allowed in TTCN-3 and on the other hand a TDL name may be a reserved keyword in TTCN-3. That is why the term TTCNname is introduced. A TTCNname of an 'Element' is the name of the 'Element' that is used in the TTCN-3 code.

A TTCNname may contain a part that is derived from the TDL name with some prefixes and/or postfixes determined by a naming convention used in the TTCN-3 code.

The present document does not specify how a TTCNname is generated from a TDL name. Neither the method how the TDL names are converted to valid TTCN-3 names nor the naming convention to be used in the TTCN-3 code, however the present document recommends a naming convention. The basic assumption of the recommended TTCNname is that it contains a part which is generated from the TDL name and it may be extended by some prefix(es) and/or postfix(es).

NOTE 1: The naming convention used in the present document is only a recommendation, in a concrete tool or implementation a different one may be used.

NOTE 2: In the following clauses the function TTCNname() will be used to get the TTCNname of the corresponding 'Element'.

5.2.2 NamedElement

This is an abstract metaclass, therefore no mapping is defined.

5.2.3 PackageableElement

This is an abstract metaclass, therefore no mapping is defined.

5.2.4 Package

A 'Package' shall be mapped to a module.

```
module <<self.TTCNname()>> {
}

```

For all import: as defined in clause 5.2.5.

NOTE: In TTCN-3 a module cannot contain another module, therefore a contained 'Package' will also be mapped to a "standalone" module. If information about the 'Package' structure needs to be kept in TTCN-3, then use a suitable naming convention.

5.2.5 ElementImport

The 'ElementImport' shall be mapped to **import** statement(s).

If the 'importedElement' is empty then an **import ... all** statement shall be used:

```
import from <<self.importedPackage.TTCNname()>> all;

```

otherwise for all the 'importedElement' a selected **import** statement shall be used:

```
<< importedElement.collect(i | "import from " + " " + self.importedPackage.TTCNname() + " "
+i.getKind() + " " + i.TTCNname() ).concat(";")>>

```

NOTE: How the kind of the 'importedElement' (e.g. **type**, **template**, **const**, **function**, etc.) is determined is outside the scope of the present document. For this purpose e.g. an annotation or a naming convention can be used.

5.2.6 Comment

A 'Comment' shall be mapped to a comment:

```
/* <<self.body>> */
```

5.2.7 Annotation

If the 'key' of the 'Annotation' is the predefined 'AnnotationType' TTCN3Code, then the 'Annotation' shall be mapped to its 'value' (that is to the TTCN-3 code itself), otherwise it shall be mapped to a comment:

```
/*
ANNOTATION <<self.key.TTCNname()>>
<<self.value>>
*/
```

5.2.8 AnnotationType

'AnnotationType' shall be mapped to a comment:

```
/*
ANNOTATION TYPE <<self.TTCNname()>>
*/
```

If the 'AnnotationType' has an extension:

```
/*
ANNOTATION TYPE <<self.TTCNname()>> EXTENDS <<self.extension.extending.TTCNname()>>
*/
```

5.2.9 TestObjective

The 'TestObjective' shall be mapped to a comment:

```
/*
Test Objective <<self.name>>
Description: <<self.description>>
Objective URI: <<self.objectiveURI>>
*/
```

5.2.10 Extension

This metaclass has no dedicated mapping, it is used solely in mapping of other metaclasses.