SLOVENSKI STANDARD
SIST ES 202 784 V1.1.1:2010

01-februar-2010

Metode za preskušanje in specificiranje (MTS) - 3. različica preskušanja in zapisa krmilnih preskusov - Razširitev nabora jezikov TTCN-3: napredni parametri

Methods for Testing and Specification (MTS) - The Testing and Test Control Notation version 3 - TTCN-3 Language Extensions: Advanced Parameterization

Ta slovenski standard je istoveten z: ES 202 784 Version 1.1.1

ICS:

35.060	Jeziki, ki se uporabljajo v informacijski tehniki in tehnologiji	Languages used in information technology

SIST ES 202 784 V1.1.1:2010 **en**

ETSI ES 202 784 V1.1.1 (2009-07)

ETSI Standard

Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization

Reference
DES/MTS-00123 T3Ext_AdvParam

Keywords
conformance, testing, TTCN

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

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the 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
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

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 (http://webapp.etsi.org/IPR/home.asp).

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 relates to the multi-part ES 201 873 standard covering the Testing and Test Control Notation version 3, as identified below:

Part 1: "TTCN-3 Core Language";

Part 2: "TTCN-3 Tabular presentation Format (TFT)";

Part 3: "TTCN-3 Graphical presentation Format (GFT)";

Part 4: "TTCN-3 Operational Semantics";

Part 5: "TTCN-3 Runtime Interface (TRI)";

Part 6: "TTCN-3 Control Interface (TCI)";

Part 7: "Using ASN.1 with TTCN-3";

Part 8: "The IDL to TTCN-3 Mapping";

Part 9: "Using XML schema with TTCN-3";

Part 10: "TTCN-3 Documentation Comment Specification".

1 Scope

The present document defines the Advanced Parameterization package of TTCN-3. TTCN-3 can be used for the specification of all types of reactive system tests over a variety of communication ports. Typical areas of application are protocol testing (including mobile and Internet protocols), service testing (including supplementary services), module testing, testing of CORBA based platforms, APIs, etc. TTCN-3 is not restricted to conformance testing and can be used for many other kinds of testing including interoperability, robustness, regression, system and integration testing. The specification of test suites for physical layer protocols is outside the scope of the present document.

TTCN-3 packages are intended to define additional TTCN-3 concepts, which are not mandatory as concepts in the TTCN-3 core language, but which are optional as part of a package which is suited for dedicated applications and/or usages of TTCN-3.

This package defines:

- Value parameters of types.
- Type parameterization.

While the design of TTCN-3 package has taken into account the consistency of a combined usage of the core language with a number of packages, the concrete usages of and guidelines for this package in combination with other packages is outside the scope of the present document.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[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 201 873-4: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 4: TTCN-3 Operational Semantics".

[3] ETSI ES 201 873-5: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI)".

[4] 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)".

[5] 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".

[6] 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".

[7] ISO/IEC 9646-1: "Information technology - Open Systems Interconnection -Conformance testing methodology and framework; Part 1: General concepts".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 873-1 [1], ES 201 873-4 [2], ES 201 873-5 [3], ES 201 873-6 [4], ES 201 873-7 [5], ES 201 873-10 [6], ISO/IEC 9646-1 [7] and the following apply:

type parameterization: ability to pass a type as an actual parameter into a parameterized object via a type parameter

NOTE: This actual type parameter is added to the specification of that object and may complete it.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 201 873-1 [2], ES 20 1873-4 [2], ES 201 873-5 [3], ES 201 873-6 [4], ES 201 873-7 [5], ES 201 873-10 [6] and ISO/IEC 9646-1 [7] apply.

4 Package conformance and compatibility

The package presented in the present document is identified by the package tag:

- `"TTCN-3:2009 Advanced Parameterization"` - to be used with modules complying with the present document.

For an implementation claiming to conform to this package version, all features specified in the present document shall be implemented consistently with the requirements given in the present document and in ES 201 873-1 [1] and ES 201 873-4 [2].

The package presented in this document is compatible to:

ES 201 873-1, version 4.1.1;

ES 201 873-2, version 3.2.1;

ES 201 873-3, version 3.2.1;

ES 201 873-4, version 4.1.1;

ES 201 873-5, version 4.1.1;

ES 201 873-6, version 4.1.1;

ES 201 873-7, version 4.1.1;

ES 201 873-8, version 3.3.1;

ES 201 873-9, version 4.1.1;

ES 201 873-10, version 3.4.1.

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

5 Package concepts for the core language

5.1 Extension to ES 201 873-1, clause 4 (Introduction)

The present package adds the following essential characteristic to TTCN-3:

- type parameterization.

5.2 Extension to ES 201 873-1, clause 5 (Basic language elements)

Clause 5.2.1 Scope of formal parameters

Add the following text:

Additionally, formal type parameters can be used as types of formal value parameters, return values, **`runs on`** and **`system`** clauses, where applicable.

Clause 5.4 Parameterization

Additionally, TTCN-3 supports type parameterization.

Replace table 2 "Overview of parameterizable TTCN-3 objects" with the following table.

Table 2: Overview of parameterizable TTCN-3 objects

Keyword	Allowed kind of Parameterization	Allowed form of non-type Parameterization	Allowed types in formal non-type parameter lists
module	Value parameterization	Static at start of run-time	all basic types, all user-defined types and `address` type.
type	Value parameterization, type parameterization	Static at compile-time	all basic types, all user-defined types and `address` type.
template	Value and template parameterization, type parameterization	Dynamic at run-time	all basic types, all user-defined types, `address` type, `template`.
function	Value, template, port and timer parameterization, type parameterization	Dynamic at run-time	all basic types, all user-defined types, `address` type, `component` type, `port` type, `default`, `template` and `timer`.
altstep	Value, template, port and timer parameterization, type parameterization	Dynamic at run-time	all basic types, all user-defined types, `address` type, `component` type, `port` type, `default`, `template` and `timer`.
testcase	Value, template, port and timer parameterization, type parameterization	Dynamic at run-time	all basic types and of all user-defined types, `address` type, `template`.
signature	Value and template parameterization, type parameterization	Dynamic at run-time	all basic types, all user-defined types and `address` type, `component` type.
NOTE: Type parameterization is always static at compile-time.			

Clause 5.4.1 Formal parameters

All types in TTCN-3 may be parameterized.

Clause 5.4.1.1 Formal parameters of kind value

In addition to the existing rules, TTCN-3 supports value parameterizations as follows:

- the value parameters of user-defined types shall be in parameters
- the language element `signature` does not support *static* value parameterization.

Modify the text as follows:

Restriction a) is relaxed to:

a) Language elements which cannot be parameterized are: `const`, `var`, `timer`, `control`, ~~`record of`, `set of`, `enumerated`, `port`, `component` and sub-type definitions,~~ `group` and `import`.

Clause 5.4.1 Formal Parameters

Is extended by the following clause:

5.4.1.5 Formal parameters of kind type

Type, template, and behaviour definitions in TTCN-3 can have parameters of kind type.

Syntactical Structure

```
[ in ] [ TypeIdentifier ] TypeParIdentifier [ ":=" Type ]
```

Semantic Description

Types passed into a parameterized object can be used inside the definition of that object. This includes the usage as type of value, template, and port parameters, as type of return values, and within `runs on` and `system` clauses of behaviour definitions.

Any type parameterization shall be resolved statically.

Type parameters will be written in a separate parameterlist enclosed in angle brackets.

Parameters of type kind may have a default type, which is given by a type assigned to the parameter.

The actual parameters of a type parameter can be required to be compatible with a specific component type. This is indicated by referring to a specific component type in the formal parameter list instead of using the keyword **`type`**.

Restrictions

a) Formal type parameters shall be in parameters, which can optionally be indicated by the optional keyword **in**.

b) The default type has to be compatible with the type of the parameter. For type compatibility see [1] TTCN-3 Core Language clause 6.4. The default type shall not refer to other type parameters in the same parameter list.

c) Requiring type compatibility of the actual parameter with the formal parameter is possible for component types only.

d) External functions shall not have type parameters.

Examples

```
 // Definition of a list and a check function
type record of T MyList <in type T>;
function isElement <in type T>(in MyList<T> list, in T elem) return boolean { … }

// Definition of a protocol message
type record Data<in type PayloadType> {
  Header      hdr,
  PayloadType payload
}

// restricting the actual type parameters
// the function can create a component of a type that is an extension of CT.
type component CT { timer t_guard };
function MyFunction <in CT Comp> (in integer p) runs on CT {
  var Comp c := Comp.create;
  :
};
```


Clause 5.4.1.1 Formal Parameters of kind value

Formal parameters with default values are additionally restricted by:

Restrictions

Replace the text as follows:

e) ~~The expression of the default value has to be compatible with the type of the parameter. The expression shall not refer to elements of the component type of the optional `runs on` clause. The expression shall not refer to other parameters of the same parameter list. The expression shall not contain the invocation of functions with a `runs on` clause.~~

e) The type of a value parameter with a default value shall not be a type parameter.

Clause 5.4.1.2 Formal Parameters of kind template

Formal parameters with default templates are additionally restricted by

Restrictions

a) ~~Only `function`, `testcase`, `altstep` and `template` definitions may have formal template parameters.~~a) The type of a template parameter with a default template shall not be a type parameter.