ETSI ES 201 873-1 V4.12.1 (2020-05)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

Part 1: TTCN‑3 Core Language

**ETSI Standard**

Reference

RES/MTS-201873-1V4.12.1

Keywords

language, methodology, 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](http://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.

# 14 Declaring procedure signatures

Procedure signatures (or signatures for short) are needed for procedure-based communication. Procedure-based communication may be used for the communication within the test system, i.e. among test components, or for the communication between the test system and the SUT. In the latter case, a procedure may either be invoked in the SUT (i.e. the test system performs the call) or in the test system (i.e. the SUT performs the call).

***Syntactical Structure***

**signature** *SignatureIdentifier*

"(" { [ **in** | **inout** | **out** ] *Type* *ValueParIdentifier* [ ","] } ")"

[ ( **return** *Type* ) | **noblock** ]

[ **exception** "(" *ExceptionTypeList* ")" ]

***Semantic Description***

For all used procedures, i.e. procedures used for the communication among test components, procedures called from the SUT and procedures called from the test system, a procedure **signature** shall be defined in the TTCN‑3 module.

TTCN‑3 supports *blocking* and *non-blocking* procedure-based communication. By default, signature definitions without the **noblock** keyword are assumed to be used for blocking procedure-based communication.

Signature definitions may have parameters. Parameters shall be of data type only, i.e. of a basic type, a structured type thereof or a subtype thereof. Within a **signature** definition the parameter list may include parameter identifiers, parameter types and their direction, i.e. **in**, **out**, or **inout**. The direction **inout** and **out** indicate that these parameters are used to retrieve information from the remote procedure.

NOTE 1: The direction of the parameters is as seen by the *called* party rather than the *calling* party.

A remote procedure may return a value after its termination. The type of the return value shall be of data type only and shall be specified by means of a **return** clause in the corresponding signature definition.

Exceptions that may be raised by remote procedures are represented in TTCN‑3 as values of a specific type. Therefore templates and matching mechanisms can be used to specify or check return values of remote procedures.

NOTE 2: The conversion of exceptions generated by or sent to the SUT into the corresponding TTCN‑3 type or SUT representation is tool and system specific and therefore beyond the scope of the present document.

The exceptions are defined in the form of an exception list included in the **signature** definition. This list defines all the possible different types associated with the set of possible exceptions (the meaning of exceptions themselves will usually only be distinguished by specific values of these types).

***Restrictions***

In addition to the general static rules of TTCN‑3 given in clause 5, the following restrictions apply:

1. Signature definitions for non-blocking communication shall use the **noblock** keyword, shall only have **in** parameters and shall have no return value but may raise exceptions.
2. Signature parameters and the return type shall be of a data type.

***Examples***

**signature** MyRemoteProcOne (); // MyRemoteProcOne will be used for blocking

// procedure-based communication. It has neither

// parameters nor a return value.

**signature** MyRemoteProcTwo () **noblock**; // MyRemoteProcTwo will be used for non blocking

// procedure-based communication. It has neither

// parameters nor a return value.

**signature** MyRemoteProcThree (**in** **integer** Par1, **out** **float** Par2, **inout** **integer** Par3);

// MyRemoteProcThree will be used for blocking procedure-based communication. The procedure

// has three parameters: Par1 an in parameter of type integer, Par2 an out parameter of

// type float and Par3 an inout parameter of type integer.

**signature** MyRemoteProcFour (**in** **integer** Par1) **return** **integer**;

// MyRemoteProcFour will be used for blocking procedure-based communication. The procedure

// has the in parameter Par1 of type integer and returns a value of type integer after its

// termination

**signature** MyRemoteProcFive (**inout** **float** Par1) **return** **integer**

**exception** (ExceptionType1, ExceptionType2);

// MyRemoteProcFive will be used for blocking procedure-based communication. It returns a

// float value in the inout parameter Par1 and an integer value, or may raise exceptions of

// type ExceptionType1 or ExceptionType2

**signature** MyRemoteProcSix (**in** **integer** Par1) **noblock**

**exception** (**integer**, **float**);

// MyRemoteProcSix will be used for non-blocking procedure-based communication. In case of

// an unsuccessful termination, MyRemoteProcSix raises exceptions of type integer or float.