ETSI ES 201 873-1 V4.11.1 (2019-04)

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-201873-1v4.11.1\_Core

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 2019.

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 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.

### 6.2.10 Component types

#### 6.2.10.1 Component type definition

The component type defines which ports are associated with a component (see figure 3). The port names in a component type definition are local to that component type, i.e. another component type may have ports with the same names. Port names in the same component type definition shall all have unique names. If not stated otherwise, ports have the same semantics as constants of a port type.



Figure 3: Typical components

It is also possible to declare constants, variables, templates and timers local to a particular component type. These declarations are visible to all testcases, functions and altsteps that run on an instance of the given component type. This shall be explicitly stated using the **runs** **on** keyword (see clause 16) in the testcase, function or altstep header. Component type definitions are associated with the component instance and follow the scope rules defined in clause 5.2. Each new instance of a component type will thus have its own set of constants, variables, templates and timers as specified in the component type definition (including any initial values, if stated). Constants used in the constant expressions of type declarations for variables, constants or ports shall meet with the restrictions in clause 10, however constants used in the constant expressions of initial values for variables, constants, templates or timers do not have to obey these restrictions.

***Syntactical Structure***

**type** **component** *ComponentTypeIdentifier* "{"

 { ( *PortInstance*

 | *VarInstance*

 | *TimerInstance*

 | *ConstDef*

 | *TemplateDef* ) }

"}"

***Semantic Description***

Component type definitions specify the creation, declaration and initialization of ports and component constants, variables, templates and timers during the creation of an instance of a component type. These instances can be used as the main test component, as the test system interface or as a parallel test component. Every instance of a component type has its own new instances of the ports, constants, variables, templates and timers defined in the component type definition.

Component instances are object references and follow specific rules for this kind of values.

***Restrictions***

No specific restrictions in addition to the general static rules of TTCN‑3 given in clause 5.

***Examples***

EXAMPLE 1: Component type with port instances only

 **type component** MyPTCType

 {

 **port** MyMessagePortType pCO1, pCO4;

 **port** MyProcedurePortType pCO2;

 **port** MyAllMesssagesPortType pCO3

 }

EXAMPLE 2: Component type with variable, timer and port instance

 **type component** MyMTCType

 {

 **var** integer vc\_myLocalInteger;

 **timer** tc\_myLocalTimer;

 **port** MyMessagePortType pCO1

 }

EXAMPLE 3: Component type with port instance arrays

 **type component** MyCompType

 {

 **port** MyMessageInterfaceType pCO[3]

 **port** MyProcedureInterfaceType pCOm[3][3]

 // Defines a component type which has an array of 3 message ports and a two‑dimensional

 // array of 9 procedure ports.

 }