ETSI ES 203 022 V1.2.1 (2018-05)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

TTCN-3 extension: Advanced Matching

**ETSI Standard**

Reference

RES/MTS-203022ed121

Keywords

conformance, 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 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.

## 5.1 Dynamic Matching

A dynamic matching is a special matching mechanism. Similar to other matching mechanisms, it can be considered as a Boolean function that indicates successful matching for the value to be matched by returning the value **true** and unsuccessful matching by returning the value **false**.

Syntactical Structure

**@dynamic** (*StatementBlock | FunctionRef*)

Semantic Description

The *StatementBlock* shall return a value of type **boolean**. The value to be matched is referenced by the special keyword **value**. When applying this matching mechanism to a value, the *StatementBlock* is executed and if and only if the execution returns **true**, the dynamic matching function matches. Unsuccessful matching shall return **false**.

A dynamic matching function can only be used in the context of a template, the **value** expression inside the *StatementBlock* shall have the same type as the whole template.

The notation **@dynamic** *FunctionRef* denotes a shorthand for the special case **@dynamic { return** *FunctionRef***(value) }** where *FunctionRef* is a reference to a Boolean function with the first parameter compatible with the template's type. If the function contains more than one parameter, all parameters following the first one shall have a default value, The type of the first parameter of the referenced function determines the type context, if this template's place of usage does not provide a type context.

Restrictions

1. The dynamic matching syntax shall only be used in a typed context.
2. The *StatementBlock* shall compute a value of type **boolean**.
3. The *StatementBlock* shall be deterministic and side-effect free and follow the restrictions of clause 16.1.4 of ETSI ES 201 873-1 [1].
4. The *StatementBlock* shall not use variables that are declared outside of the *StatementBlock*.
5. The *StatementBlock* shall not use **inout** or **out** parameters.
6. Only if the dynamic matching syntax appears on the right-hand-side of a parameterized template definition, the formal **in** parameters of that definition may be referenced inside the *StatementBlock*. All other formal **in** parameters shall not be used by the *StatementBlock*.

EXAMPLE:

 type record of integer Numbers;

 template Numbers mw\_sorted := @dynamic { // value is of type Numbers

 for (var integer v\_i := 1; v\_i < lengthof(value); v\_i := v\_i + 1) {

 if (value[v\_i-1] > value[v\_i]) { return false }

 }

 return true;

 } // mw\_sorted(v\_recInt) matches all values of type Numbers

// if elements of v\_recInt do not break an ascending order

 type record Coordinate { float x, float y };

 external function fx\_distance(Coordinate p\_a, Coordinate p\_b) return float;

 template float mw\_closeTo(Coordinate p\_origin := { 0.0, 0.0 }, float p\_maxDistance := 1.0) :=

 // access to in parameters is allowed

 @dynamic { return fx\_distance(p\_origin, value) <= p\_maxDistance; };

 // mw\_closeTo(c,d) matches all values of type Coordinate

// which have maximum distance of d from Coordinate c

 external function fx\_isPrime(integer p\_x) return boolean;

 :

 p.receive(@dynamic fx\_isPrime)

 // is the same as p.receive(integer:@dynamic { return fx\_isPrime(value) })