ETSI ES 202 782 V1.4.1 (2022-06)

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1-4-1-2022-06

Reference
RES/MTS-202782 ed141RealtPer

Keywords
performance, real time, 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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

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

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

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 2022.
All rights reserved.

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/1c4b96f1-dc30-4134-aa96-...

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1-4-1-2022-06

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Foreword

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

https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-

The present document relates to the multi-part standard ETSI ES 201 873 covering the Testing and Test Control Notation version 3, as identified in ETSI ES 201 873-1 [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 defines the real time and performance testing support 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 OMG 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.

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

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 reference 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 201 873-1 (V4.6.1): "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 (V4.4.1): "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 (V4.6.1): "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 (V4.6.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI)".

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

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 reference 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.

iTeh Standards (standards.iteh.ai) ETSI ES 202 782 V1.4.1 (2022-06) https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1-4-1-2022-06

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 ES 201 873-7 (V4.5.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 7: Using ASN.1 with TTCN-3".

[i.2] ETSI ES 201 873-8 (V4.5.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 8: The IDL to TTCN-3 Mapping".

[i.3] ETSI ES 201 873-9 (V4.5.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3".

[i.4] ETSI ES 201 873-10 (V4.5.1): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 10: TTCN-3 Documentation Comment Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

3.2 Symbols

Void.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

3.3 Abbreviations

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

ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/[illegible]-9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1-4-1-2022-06

4 Package conformance and compatibility

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

`"TTCN-3:2014 Real Time and Performance Testing"` - 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 ETSI ES 201 873-1 [1], ETSI ES 201 873-4 [2], ETSI ES 201 873-5 [3] and ETSI ES 201 873-6 [4].

The package presented in the present document is compatible with:

- ETSI ES 201 873-1 (V4.6.1) [1]
- ETSI ES 201 873-4 (V4.4.1) [2]
- ETSI ES 201 873-5 (V4.6.1) [3]
- ETSI ES 201 873-6 (V4.6.1) [4]
- ETSI ES 201 873-7 (V4.5.1) [i.1]
- ETSI ES 201 873-8 (V4.5.1) [i.2]
- ETSI ES 201 873-9 (V4.5.1) [i.3]

- ETSI ES 201 873-10 (V4.5.1) [i.4]

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. The present document is also compatible with the versions V4.2.1, V4.3.1, V4.4.1 and V4.5.1 of the above documents.

5 Package concepts for the core language

5.0 General

Real-time systems have to respect special requirements for timing. Often functional requirements are directly connected to the timing of the messages and procedure calls. Thus, checking the message values and the message order is not sufficient here. A test component shall be able to check whether a message has been received in time and shall be able to control the timing for the stimulation.

Thus, a test language has to provide means to measure time, to specify time points and time spans, to control the timing of the stimulation, and to calculate and compare time values. Moreover the test execution engine has to ensure that the specified actions (time measurement, timed stimulation) are executed correctly with respect to the required precision.

To fulfil the requirements for testing real time system the following TTCN-3 core language extensions have been defined:

- A test system wide available test system clock, that allows the measurement of time during test case execution.
- Means to directly and precisely access the time points of the relevant interaction events between the test system and the system under test.

Real-time measurements at ports require additional resources (e.g. functionality that monitor ports and collect timestamps that describe the reception time of messages, calls, replies or exceptions) that may slow down the test execution. In order to avoid unnecessary delays at ports, such resources may only be provided when needed. An additional **`real-time`** clause for ports shall indicate the need for real-time measurement at a port.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1-4-1-2022-06

5.1 The test system clock

5.1.0 General

In RT TTCN-3 time progress is measured with a test system clock. The clock is initialized (set to 0.0) at the beginning of each test case execution and is available during the complete test run in each component. The clock values are represented as float values. The system clock and the already available TTCN-3 timer mechanisms are synchronized with respect to time progress.

5.1.1 Accessing the current test system time

The current value of the test system clock by means of the symbol **`now`**. The **`now`** symbol is used as a TTCN-3 expression that yields the current test system clock value in seconds. The test system clock value is represented by means of a **`float`** number. The symbol **`now`** can be applied in each expression inside of testcase definitions and function definitions. It is not allowed for the TTCN-3 control part and in guard conditions of alt branches.

EXAMPLE 1:

```
// Use of now to retrieve the actual time
var float myTimePoint := now;
```

EXAMPLE 2:

```
// Use of now to retrieve the send time of a message
var float sendTimePoint;
// ...
p.send(m);
sendTimePoint:= now;
```

EXAMPLE 3:

```
// Measuring time progress
var float startTime;
startTime:= now;
p.send(m1);
// ...
p.receive(m2);
if(now-startTime >= 10.0){...};
```

Syntactical Structure

```
OpCall ::= ConfigurationOps | VerdictOps | TimerOps | TestcaseInstance |
                              FunctionInstance | TemplateOps | ActivateOp | NowOperation
NowOperation ::=  NowKeyword
NowKeyword ::=  "now"
```

5.1.2 The precision of the system time

The requirements on the overall precision of the test system clock can be specified by means of the stepsize annotation. The stepsize annotation is allowed for modules only and can be used to state the minimal necessary precision for time measurement provided by the test system clock. The precision is defined by means of a charstring value that represents a decimal number which states the smallest necessary time distance in seconds that is measurable by the test system clock. A concrete test system has to fulfil the requirements given by the stepsize annotation to be adequate for the execution of the respective test case definitions. When a test system is not adequate for the test case execution the user shall be informed, at least test run shall end with an error verdict.

EXAMPLE:

```
// specifies the requirement on a necessary precision of a millisecond
module myModule{
…
} with {stepsize “0.001”};
```

iTeh STANDARD PREVIEW
(standards.iteh.ai)

In case of module imports with different stepsize annotation the test system has to respect the stepsize annotation with the highest precision.

ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-1e4b05f4158f/etsi-es-202-782-v1.4.1-2022-06

5.2 Communication port types for real-time measurements

This package extends the port type definition of message-based and procedure-based ports with a **`realtime`** clause. Ports facilitate communication between test components and between test components and the test system interface.

Only instances of ports with a realtime clause shall be used for real-time measurements. This means, the redirection operator **`-> timestamp`** shall only be used by receiving operations (i.e. the operations **`receive`**, **`trigger`**, **`getcall`**, **`getreply`** and **`catch`**) applied to ports with a **`realtime`** clause.

Syntactical Structure

Message-based port:

```
type port PortTypeIdentifier message [realtime] "{"
    { ( in | out | inout ) { MessageType [ "," ] }+ ";" }
"}"
```

Procedure-based port:

```
type port PortTypeIdentifier procedure [realtime] "{"
    { ( in | out | inout ) { Signature [ "," ] }+ ";" }
"}"
```

5.3 Measuring timing information for dedicated incoming communication events

5.3.0 General

Testing real time systems requires exact timing information that relates directly to the communication (reception and distribution of messages and procedure calls) between the test system and the system under test. The timing information that can be obtained by the **now** symbol or the TTCN-3 timer construct is related to the logical structure of the test program, thus it allows the measurement on TTCN-3 statement level. Time measurement on TTCN-3 statement level may be affected by blocked queues, decoding and matching procedures. It is not exact with respect to the real timing of the reception and disposal of messages and procedure calls at the interface between the test system and the SUT.

RT TTCN-3 introduces a mechanism to store the arrival time of messages, procedure calls at system adapter level. The time points of message reception are automatically registered by the system adapter, communicated to the test executable and stored with the message. The timing information can be retrieved directly at the communication statements by means of the redirection operator **-> timestamp**.

The existing redirections for getcall, getreply, receive, trigger, catch, and check operations are extended by an optional clause **timestamp.** A redirect specification of the form:

```
-> timestamp VariableRef
```

specifies the redirection of the time point, which has been measured at message, procedure call, reply or exception arrival to a given float variable. The redirection is processed when the respective communication statement matches.

Restrictions

iTeh STANDARD PREVIEW

The redirection operator **-> timestamp** shall only be used by receiving operations (i.e. the operations **receive**, **trigger**, **getcall**, **getreply** and **catch**) applied to ports with a **realtime** clause.

(standards.iteh.ai)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-

5.3.1 Obtain the reception time for messages with the receive statement

The existing redirections for receive are extended by an optional clause "**timestamp** VariableRef". A receive statement that holds a timestamp clause and that is executed successfully (i.e. it matches a message) allocates the given variable with the reception time of the matched message.

EXAMPLE 1:

```
p.receive(t)-> timestamp myTime;
// yields the reception time of a message
if(myTime>MAX){setverdict(fail);}
```

EXAMPLE 2:

```
interleave{
    [ ] FrontOut.receive(ON) -> timestamp f_actv{
            if(f_actv>MAX){setverdict(fail);}
        };
    [ ] RearOut.receive(ON) -> timestamp r_actv{
            if(r_actv>MAX){setverdict(fail);}
        };
}
```

Syntactical Structure

```
( Port | any port ) "." receive ["(" TemplateInstance ")"] [ from AddressRef ]
[ -> [ value  VariableRef ] [ sender VariableRef ] [ timestamp VariableRef] ]
```

5.3.2 Obtain the reception time for messages with the trigger statement

The existing redirections for trigger are extended by an optional clause "**timestamp** VariableRef". A trigger statement that holds a timestamp clause and that is executed successfully (i.e. it matches a message) allocates the given variable with the reception time of the matched message.

EXAMPLE 1:

```
p.trigger(t)-> timestamp myTime;
// yields the reception time of a message
if(myTime>MAX){setverdict(fail);}
```

EXAMPLE 2:

```
interleave{
    [ ] FrontOut.trigger(ON) -> timestamp f_actv{
            if(f_actv>MAX){setverdict(fail);}
        };

    [ ] RearOut.trigger(ON) -> timestamp r_actv{
            if(r_actv>MAX){setverdict(fail);}
        };
}
```

Syntactical Structure

```
( Port | any port ) "." trigger [ "(" TemplateInstance ")" ] [ from AddressRef ]
[ -> [ value  VariableRef ] [ sender VariableRef ] ] [ timestamp VariableRef] ]
```

5.3.3 Obtain the reception time for procedure calls with getcall statement

The existing redirections for getcall are extended by an optional clause "**timestamp** VariableRef". A getcall statement that holds a timestamp clause and that is executed successfully (i.e. it matches an incoming call) allocates the given variable with the reception time of the matched message.

EXAMPLE 1:

```
p.getcall(proc: {m})-> timestamp myTime;
// yields the reception time of the message call matched by m
if(myTime>MAX){setverdict(fail);}
```

EXAMPLE 2:

```
alt{
    [ ] p.getcall(proc: {m1})-> timestamp f_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
    [ ] p.getcall(proc: {m2})-> timestamp r_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
}
```

Syntactical Structure

```
( Port | any port ) "." getcall  [ "(" TemplateInstance ")" ] [ from AddressRef ]
 [ "->"  [ param "(" { VariableRef ":=" ParameterIdentifier ) "," }  |
                        { VariableRef | NotUsedSymbol ) "," }
                    ")" ]
           [ sender VariableRef ]
           [ timestamp VariableRef]
]
```

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI ES 202 782 V1.4.1 (2022-06)
https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-[illegible]58f/etsi-es-202-782-v1.4.1-2022-06

5.3.4 Obtain the reception time for procedure replies with the getreply statement

The existing redirections for getreply are extended by an optional clause "**timestamp** VariableRef". A getreply statement that holds a timestamp clause and that is executed successfully (i.e. it matches an incoming procedure reply) allocates the given variable with the reception time of the matched message.

EXAMPLE 1:

```
p.getreply(proc: {m})-> timestamp myTime;
// yields the reception time of the message call matched by m
if(myTime>MAX){setverdict(fail);}
```

EXAMPLE 2:

```
p.call(proc: {_message:= m},20.0){
    [ ] p.getreply(proc: {m1})-> timestamp f_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
    [ ] p.getreply(proc: {m2})-> timestamp r_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
}
```

Syntactical Structure

```
( Port | any port ) "." getreply [ "(" TemplateInstance  [ value TemplateInstance ] ")" ] [ from
AddressRef ]
[ "->" [ value VariableRef ]
         [  param "(" { VariableRef ":=" ParameterIdentifier ) "," }  |
                      { VariableRef | NotUsedSymbol ) "," }
                  ")" ]
         [ sender VariableRef ]
         [ timestamp VariableRef]
]
```

iTeh STANDARD PREVIEW
(standards.iteh.ai)

5.3.5 Obtain the reception time for exceptions with the catch statement

https://standards.iteh.ai/catalog/standards/sist/46d9a9f1-dc30-4134-aa96-

The existing redirections for **catch** are extended by an optional clause "**timestamp** VariableRef". A catch statement that holds a timestamp clause and that is executed successfully (i.e. it matches an incoming exception) allocates the given variable with the reception time of the matched message.

EXAMPLE 1:

```
p.catch(timeout)-> timestamp myTime;
// yields the reception time of the message call matched by m
if(myTime>MAX){setverdict(fail);}
```

EXAMPLE 2:

```
p.call(proc: {_message:= m},20.0){
    [ ] p.getreply(proc: {m1})-> timestamp f_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
    [ ] p.catch(*)-> timestamp r_actv {
            if(f_actv>MAX){setverdict(fail);}
        };
}
```

Syntactical Structure

```
( Port | any port ) "." catch [ "(" ( Signature "," TemplateInstance ) | TimeoutKeyword   ")" ] [
from AddressRef ]
[ "->" [ value VariableRef
         [ sender VariableRef ]
         [ timestamp VariableRef]
]
```