ETSI ES 202 789 V1.6.1 (2022-04)

ETSI STANDARD

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

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.

Contents

Intellectual Property Rights 4

Foreword...4

Modal verbs terminology..4

1   Scope 5

2   References 5
2.1   Normative references 5
2.2   Informative references 6

3   Definition of terms, symbols and abbreviations..6
3.1   Terms 6
3.2   Symbols 6
3.3   Abbreviations 7

4   Package conformance and compatibility..7

5   Package concepts for the core language 7

6   Package semantics..7

7   TRI extensions for the package 8
7.0   Introduction 8
7.1   Changes to clause 5.2 of ETSI ES 201 873-5, Error handling 8
7.2   Changes to clause 5.5.2 of ETSI ES 201 873-5, Connection handling operations 9
7.3   Changes to clause 5.5.3 of ETSI ES 201 873-5, Message based communication operations............................9
7.4   Addition to clause 5.5.3 of ETSI ES 201 873-5, Message based communication operations 11
7.5   Changes to clause 5.5.4 of ETSI ES 201 873-5, Procedure based communication operations 11
7.5A   Changes to clause 5.5.5 of ETSI ES 201 873-5, Miscellaneous operations 18
7.5B   Addition to clause 5.5.5 of ETSI ES 201 873-5, Miscellaneous operations 18
7.6   Changes to clause 5.6.3 of ETSI ES 201 873-5, Miscellaneous operations 18
7.7   Changes to clause 6 of ETSI ES 201 873-5, Java language mapping 19
7.8   Changes to clause 7 of ETSI ES 201 873-5, C language mapping 22
7.9   Changes to clause 8 of ETSI ES 201 873-5, C++ language mapping 24
7.10   Changes to clause 9 of ETSI ES 201 873-5, C# language mapping..27

8   TCI extensions for the package 28

Annex A (informative):   Bibliography 29

History 30

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.

Foreword

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

The use of strike through (deleted text) highlights the differences between base document and extended documents.

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 Extended TRI 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 or in its interfaces TRI and TCI, but which are optional as part of a package which is suited for dedicated applications and/or usages of TTCN-3.

This package defines a more efficient handling of software values by a version of TRI, that does not use binary encoded messages for the communication with the SUT, but uses the values as they are; meaning e.g. that software objects or serialized data can be passed directly between the SUT and the TE.

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 referenced 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: "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]   Recommendation ITU-T X.290: "OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications - General concepts".

NOTE: The corresponding ISO/IEC standard is 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 referenced 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.

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]   Void.

[i.2]   Void.

[i.3]   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".

[i.4]   ETSI ES 201 873-8: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 8: The IDL to TTCN-3 Mapping".

[i.5]   ETSI ES 201 873-9: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3".

[i.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".

[i.7]   ETSI ES 202 781: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Configuration and Deployment Support".

[i.8]   ETSI ES 202 784: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Advanced Parameterization".

[i.9]   ETSI ES 202 785: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Behaviour Types".

[i.10]   ETSI ES 202 782: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: TTCN-3 Performance and Real Time Testing".

[i.11]   ETSI ES 202 786: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Language Extensions: Support of interfaces with continuous signals".

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 Recommendation ITU-T X.290 [5] apply.

3.2 Symbols

Void.

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], Recommendation ITU-T X.290 [5] and the following apply:

XTRI   eXtended TRI

4   Package conformance and compatibility

The package has no package tag as the choice to use TRI and/or XTRI affects the test adaptor only, but not the test specifications in TTCN-3.

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, ETSI ES 201 873-1 [1] and ETSI ES 201 873-4 [2].

The package presented in the present document is compatible to:

ETSI ES 201 873-1 [1] (V4.5.1)

ETSI ES 201 873-4 [2] (V4.4.1)

ETSI ES 201 873-6 [4] (V4.5.1)

ETSI ES 201 873-7 [i.3] (V4.5.1)

ETSI ES 201 873-8 [i.4] (V4.5.1)

ETSI ES 201 873-9 [i.5] (V4.5.1)

ETSI ES 201 873-10 [i.6] (V4.5.1)

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

The package defined in the present document is also compatible to:

ETSI ES 202 784 [i.8] (V1.3.1)

ETSI ES 202 781 [i.7] (V1.2.1)

ETSI ES 202 782 [i.10] (V1.2.1)

ETSI ES 202 785 [i.9] (V1.3.1)

ETSI ES 202 786 [i.11] (V1.2.1)

and can be used together with those packages.

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

5   Package concepts for the core language

Not applicable.

6   Package semantics

Not applicable.

7 TRI extensions for the package

7.0 Introduction

Historically, TTCN has been used to test communication protocols which typically use encoded messages. This has been reflected in the TRI SA and TCI CD design of TTCN-3 by encoding and decoding messages to and from bitstrings. However, TTCN-3 also supports signature-based communication for which the transformation of objects into bitstrings and vice versa is cumbersome. Furthermore, some protocols use also structured messages for which the bitstring encoding is not helpful.

Therefore, an alternative API is being defined in this extension package of TTCN-3 along which TTCN-3 values can be directly passed to/from the SUT. It is defined by redefining the operations in TRI SA and PA as follows.

7.1 Changes to clause 5.2 of ETSI ES 201 873-5, Error handling

The SA or PA can in addition provide notifications about unrecoverable error situations by use of the operations xtriSAErrorReq and xtriPAErrorReq, respectively.

5.2.1 triSAErrorReq → xtriSAErrorReq

Signature	`void xtriSAErrorReq(in string message, in any cause)`	
In Parameters	`message`	A string value, i.e. the error phrase describing the problem.
	`cause`	(Optional) cause of the problem.
Return Value	`void`	
Constraint	Shall be called whenever an error situation has occurred in the SA with the exception of errors occurring when processing SA calls initiated by the TE. These errors are reported in the operation return. The optional cause parameter can be used to provide information in addition to the error phrase in message.	
Effect	The TE will be notified about an unrecoverable error situation within the SA and may forward the error indication to the test management.	

5.2.2 triPAErrorReq → xtriPAErrorReq

Signature	`void xtriPAErrorReq(in string message, in any cause)`	
In Parameters	`message`	A string value, i.e. the error phrase describing the problem.
	`cause`	(Optional) cause of the problem.
Return Value	`Void`	
Constraint	Shall be called whenever an error situation has occurred in the PA with the exception of errors occurring when processing PA calls initiated by the TE. These errors are reported in the operation return. The optional cause parameter can be used to provide information in addition to the error phrase in message.	
Effect	The TE will be notified about an unrecoverable error situation within the PA and may forward the error indication to the test management.	

7.2 Changes to clause 5.5.2 of ETSI ES 201 873-5, Connection handling operations

5.5.2.3 triMapParam → xtriMapParam

Signature	`TriStatusType xtriMap(in TriPortIdType compPortId,`
	`                       in TriPortIdType tsiPortId,`
	`                       in TciParameterListType paramList)`

In Parameters	`compPortId` identifier of the test component port to be mapped `tsiPortId` identifier of the test system interface port to be mapped `paramList` parameters of the parameterized map
Out Parameters	n.a.
Return Value	The return status of the `triMap` operation. The return status indicates the local success (***TRI_OK***) or failure (***TRI_Error***) of the operation.
Constraints	This operation is called by the TE when it executes a TTCN-3 map operation.
Effect	The SA can establish a dynamic connection to the SUT for the referenced TSI port. The `triMap` operation returns ***TRI_Error*** in case a connection could not be established successfully, ***TRI_OK*** otherwise. The operation should return ***TRI_OK*** in case no dynamic connection needs to be established by the test system.

5.5.2.5 triUnmapParam → xtriUnmapParam

Signature	`TriStatusType xtriUnmap(in TriPortIdType compPortId,`
	`                         in TriPortIdType tsiPortId,`
	`                         in TciParameterListType paramList)`

In Parameters	`compPortId` identifier of the test component port to be unmapped `tsiPortId` identifier of the test system interface port to be unmapped `paramList` parameters of the parameterized map
Out Parameters	n.a.
Return Value	The return status of the `triUnmap` operation. The return status indicates the local success (***TRI_OK***) or failure (***TRI_Error***) of the operation.
Constraints	This operation is called by the TE when it executes any TTCN-3 unmap operation.
Effect	The SA shall close a dynamic connection to the SUT for the referenced TSI port. The `triUnmap` operation returns ***TRI_Error*** in case a connection could not be closed successfully or no such connection has been established previously, ***TRI_OK*** otherwise. The operation should return ***TRI_OK*** in case no dynamic connections have to be closed by the test system.

7.3 Changes to clause 5.5.3 of ETSI ES 201 873-5, Message based communication operations

5.5.3.1 triSend → xtriSend

Signature	`TriStatusType xtriSend(in TriComponentIdType componentId,`
	`                        in TriPortIdType tsiPortId,`
	`                        in Value SUTaddress,`
	`                        in Value sendMessage)`

In Parameters	`componentId` identifier of the sending test component `tsiPortId` identifier of the test system interface port via which the message is sent to the SUT Adaptor `SUTaddress` (optional) destination address value within the SUT `sendMessage` the value to be sent
Out Parameters	n.a.
Return Value	The return status of the `triSend` operation. The return status indicates the local success (***TRI_OK***) or failure (***TRI_Error***) of the operation.

Constraints	This operation is called by the TE when it executes a TTCN-3 unicast send operation on a component port, which has been mapped to a TSI port. This operation is called by the TE for all TTCN-3 send operations if no system component has been specified for a test case, i.e. only a MTC test component is created for a test case.
Effect	The SA can send the message to the SUT. The `triSend` operation returns **TRI_OK** in case it has been completed successfully. Otherwise **TRI_Error** shall be returned. Notice that the return value **TRI_OK** does not imply that the SUT has received `sendMessage`.

5.5.3.2   triSendBC → xtriSendBC

Signature	`TriStatusType xtriSendBC(in TriComponentIdType componentId,`
	`                          in TriPortIdType tsiPortId,`
	`                          in Value sendMessage)`

In Parameters	`componentId` identifier of the sending test component `tsiPortId` identifier of the test system interface port via which the message is sent to the SUT Adaptor `sendMessage` the value to be sent
Out Parameters	n.a.
Return Value	The return status of the `triSendBC` operation. The return status indicates the local success (**TRI_OK**) or failure (**TRI_Error**) of the operation.
Constraints	This operation is called by the TE when it executes a TTCN-3 broadcast send operation on a component port, which has been mapped to a TSI port. This operation is called by the TE for all TTCN-3 send operations if no system component has been specified for a test case, i.e. only a MTC test component is created for a test case.
Effect	The SA can broadcast the message to the SUT. The `triSendBC` operation returns **TRI_OK** in case it has been completed successfully. Otherwise **TRI_Error** shall be returned. Notice that the return value **TRI_OK** does not imply that the SUT has received `sendMessage`.

5.5.3.3   triSendMC → xtriSendMC

Signature	`TriStatusType xtriSendMC(in TriComponentIdType componentId,`
	`                          in TriPortIdType tsiPortId,`
	`                          in TciValueList SUTaddresses,`
	`                          in Value sendMessage)`

In Parameters	`componentId` identifier of the sending test component `tsiPortId` identifier of the test system interface port via which the message is sent to the SUT Adaptor `SUTaddresses` destination address values within the SUT `sendMessage` the values to be sent
Out Parameters	n.a.
Return Value	The return status of the `triSendMC` operation. The return status indicates the local success (**TRI_OK**) or failure (**TRI_Error**) of the operation.
Constraints	This operation is called by the TE when it executes a TTCN-3 multicast send operation on a component port, which has been mapped to a TSI port. This operation is called by the TE for all TTCN-3 send operations if no system component has been specified for a test case, i.e. only a MTC test component is created for a test case.
Effect	The SA can multicast the message to the SUT. The `triSendMC` operation returns **TRI_OK** in case it has been completed successfully. Otherwise **TRI_Error** shall be returned. Notice that the return value **TRI_OK** does not imply that the SUT has received `sendMessage`.

5.5.3.4 triEnqueueMsg → xtriEnqueueMsg

Signature	`void xtriEnqueueMsg(in TriPortIdType tsiPortId,`
	`                     in any SUTaddress,`
	`                     in TriComponentIdType componentId,`
	`                     in any receivedMessage)`	

In Parameters	`tsiPortId`	identifier of the test system interface port via which the message is enqueued by the SUT Adaptor
	`SUTaddress`	(optional) source address value within the SUT
	`componentId`	identifier of the receiving test component
	`receivedMessage`	the received value
Out Parameters	n.a.	
Return Value	void	
Constraints	This operation is called by the SA after it has received a message from the SUT. It can only be used when tsiPortId has been either previously mapped to a port of componentId or has been referenced in the previous triExecuteTestCase statement.	
Effect	This operation shall pass the message to the TE indicating the component componentId to which the TSI port tsiPortId is mapped. The decoding of receivedMessage has to be done in the TE.	

7.4 Addition to clause 5.5.3 of ETSI ES 201 873-5, Message based communication operations

In order to interpret unknown values along a type hypothesis, an additional xtriConvert operation is defined. It can be used in all cases where the type of the incoming value is not known. Note that typically the value type is known in procedure-based communication and sometimes in message-based communication.

5.5.3.5 xtriConvert

Signature	`Value xtriConvert(in any value, in Type typeHypothesis)`	
In Parameters	`value`	the value to be converted
	`typeHypothesis`	the type hypothesis
Out Parameters	n.a.	
Return Value	Returns the converted value, if the value is a compatible type as the `typeHypothesis`, else the distinct value `null`.	
Constraints	This operation shall be called whenever the TE has to convert a value. The TE might convert immediately after reception of the value, or might for performance considerations postpone the conversion until the actual access to the value.	
Effect	This operation converts a `value` and returns a value according to the type hypothesis if it matches. The `typeHypothesis` determines whether the value can be converted. If not, the distinct `null` value shall be returned.	

7.5 Changes to clause 5.5.4 of ETSI ES 201 873-5, Procedure based communication operations

5.5.4.1 triCall → xtriCall

Signature	`TriStatusType xtriCall(in TriComponentIdType componentId,`
	`                        in TriPortIdType tsiPortId,`
	`                        in Value SUTaddress,`
	`                        in TriSignatureIdType signatureId,`
	`                        in TciParameterListType parameterList)`

In Parameters	`componentId`	identifier of the test component issuing the procedure call
	`tsiPortId` sent	identifier of the test system interface port via which the procedure call is to the SUT Adaptor
	`SUTaddress`	(optional) destination address within the SUT
	`signatureId`	identifier of the signature of the procedure call
	`parameterList`	a list of parameters which are part of the indicated signature. The parameters in `parameterList` are ordered as they appear in the TTCN-3 signature declaration
Out Parameters	n.a.	
Return Value	The return status of the `triCall` operation. The return status indicates the local success (***TRI_OK***) or failure (***TRI_Error***) of the operation.	
Constraints	This operation is called by the TE when it executes a TTCN-3 unicast call operation on a component port, which has been mapped to a TSI port. This operation is called by the TE for all TTCN-3 call operations if no system component has been specified for a test case, i.e. only a MTC test component is created for a test case. The procedure parameters are the parameters specified in the TTCN-3 signature template.	
Effect	On invocation of this operation the SA can initiate the procedure call corresponding to the signature identifier `signatureId` and the TSI port `tsiPortId`. The `triCall` operation shall return without waiting for the return of the issued procedure call (see note). This TRI operation returns *TRI_OK* on successful initiation of the procedure call, *TRI_Error* otherwise. No error shall be indicated by the SA in case the value of any *out* parameter is non-null. Notice that the return value of this TRI operation does not make any statement about the success or failure of the procedure call. Note that an optional timeout value, which can be specified in the TTCN-3 ATS for a call operation, is *not* included in the `triCall` operation signature. The TE is responsible to address this issue by starting a timer for the TTCN-3 call operation in the PA with a separate TRI operation call, i.e. `triStartTimer`.	
NOTE:	This might be achieved for example by spawning a new thread or process. This handling of this procedure call is, however, dependent on implementation of the TE.	

5.5.4.2 triCallBC → xtriCallBC

Signature	`TriStatusType xtriCallBC(in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriSignatureIdType signatureId,`
	`in TciParameterListType parameterList)`	

In Parameters	`componentId`	identifier of the test component issuing the procedure call
	`tsiPortId`	identifier of the test system interface port via which the procedure call is sent to the SUT Adaptor
	`signatureId`	identifier of the signature of the procedure call
	`parameterList`	a list of parameters which are part of the indicated signature. The parameters in `parameterList` are ordered as they appear in the TTCN-3 signature declaration
Out Parameters	n.a.	
Return Value	The return status of the `triCallBC` operation. The return status indicates the local success (***TRI_OK***) or failure (***TRI_Error***) of the operation.	
Constraints	This operation is called by the TE when it executes a TTCN-3 broadcast call operation on a component port, which has been mapped to a TSI port. This operation is called by the TE for all TTCN-3 call operations if no system component has been specified for a test case, i.e. only a MTC test component is created for a test case. The procedure parameters are the parameters specified in the TTCN-3 signature template.	
Effect	On invocation of this operation the SA can initiate and broadcast the procedure call corresponding to the signature identifier `signatureId` and the TSI port `tsiPortId`. The `triCallBC` operation shall return without waiting for the return of the issued procedure call (see note). This TRI operation returns *TRI_OK* on successful initiation of the procedure call, *TRI_Error* otherwise. No error shall be indicated by the SA in case the value of any *out* parameter is non-null. Notice that the return value of this TRI operation does not make any statement about the success or failure of the procedure call. Note that an optional timeout value, which can be specified in the TTCN-3 ATS for a call operation, is *not* included in the `triCallBC` operation signature. The TE is responsible to address this issue by starting a timer for the TTCN-3 call operation in the PA with a separate TRI operation call, i.e. `triStartTimer`.	
NOTE:	This might be achieved for example by spawning a new thread or process. This handling of this procedure call is, however, dependent on implementation of the TE.	