



TCI /TRI Quick Reference Card

- PDF has links to TTCN-3 online Standards and browseable API -

For TTCN-3 edition 4.9.1 (2017).

Designed and edited by [Axel Rennoch](#) and [Claude Desroches](#)

Contents

TCI/TRI interface overview 2

1) TTCN-3 Control interface (CD)

A.1	TCI-CD interface	2
B.1	JAVA mapping	2

2) TTCN-3 Runtime interface

A.2a	triPlatform interface	4
A.2b	triCommunication interface	4
B.2	JAVA mapping	5

3) TTCN-3 Logging interface (TLI)

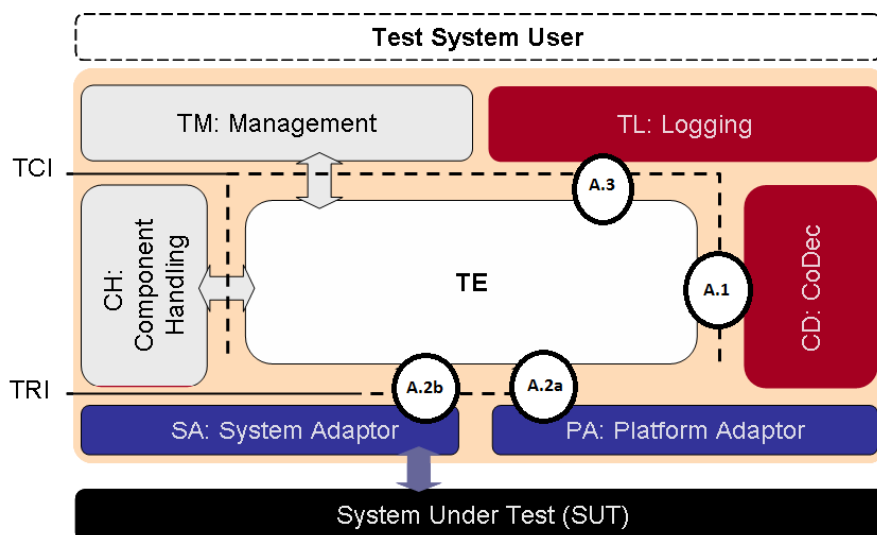
A.3	TCI-TL interface	6
B.3	TCI-TL types (W3C XML mapping)	7

Document overview:

- [1] [ES 201 873-1](#) (2017-05) TTCN-3 part 1 (edition 4.9.1): *Core Language (CL)* [\[ITU-T Z.161\]](#)
- [2] [ES 201 873-2](#) (2007-02) TTCN-3 part 2 (edition 3.2.1): *Tabular Presentation format (TFT)* (historical - not maintained!) [\[ITU-T Z.162\]](#)
- [3] [ES 201 873-3](#) (2007-02) TTCN-3 part 3 (edition 3.2.1): *Graphical Presentation format (GFT)* [\[ITU-T Z.163\]](#)
- [4] [ES 201 873-4](#) (2017-05) TTCN-3 part 4 (edition 4.6.1): *Operational Semantics (OS)* [\[ITU-T Z.164\]](#)
- [5] [ES 201 873-5](#) (2017-05) TTCN-3 part 5 (edition 4.8.1): *TTCN-3 Runtime Interface (TRI)* [\[ITU-T Z.165\]](#)
- [6] [ES 201 873-6](#) (2017-05) TTCN-3 part 6 (edition 4.9.1): *TTCN-3 Control Interface (TCI)* [\[ITU-T Z.166\]](#)
- [7] [ES 201 873-7](#) (2017-05) TTCN-3 part 7 (edition 4.6.1): *Using ASN.1 with TTCN-3* [\[ITU-T Z.167\]](#)
- [8] [ES 201 873-8](#) (2015-06) TTCN-3 part 8 (edition 4.6.1): *The IDL to TTCN-3 Mapping* [\[ITU-T Z.168\]](#)
- [9] [ES 201 873-9](#) (2017-05) TTCN-3 part 9 (edition 4.8.1): *Using XML schema with TTCN-3* [\[ITU-T Z.169\]](#)
- [10] [ES 201 873-10](#) (2013-04) TTCN-3 part 10 (edition 4.5.1): *TTCN-3 Documentation Comment Specification* [\[ITU-T Z.170\]](#)
- [11] [ES 201 873-11](#) (2017-06) TTCN-3 part 11 (edition 4.7.1): *Using JSON with TTCN-3* **NEW**

- [12] [ES 202 781](#) (2017-05) TTCN-3 Language Extensions (version 1.5.1): *Configuration and Deployment Support* [\[ITU-T Z.161.2\]](#)
- [13] [ES 202 782](#) (2015-06) TTCN-3 Language Extensions (version 1.3.1): *TTCN-3 Performance and Real Time Testing* [\[ITU-T Z.161.5\]](#)
- [14] [ES 202 784](#) (2017-04) TTCN-3 Language Extensions (version 1.6.1): *Advanced Parameterization* [\[ITU-T Z.161.3\]](#)
- [15] [ES 202 785](#) (2015-06) TTCN-3 Language Extensions (version 1.4.1): *Behaviour Types* [\[ITU-T Z.161.4\]](#)
- [16] [ES 202 786](#) (2017-05) TTCN-3 Language Extensions (version 1.4.1): *Support of interfaces with continuous signals* [\[ITU-T Z.161.1\]](#)
- [17] [ES 202 789](#) (2015-06) TTCN-3 Language Extensions (version 1.4.1): *Extended TRI* [\[ITU-T Z.165.1\]](#)
- [18] [ES 203 022](#) (2017-07) TTCN-3 Language Extensions (version 1.1.1): *Advanced Matching* **NEW**
- [19] [TS 102 995](#) (2010-11) *Proforma for TTCN-3 reference test suite* (version 1.1.1)

TCI / TRI interface overview



A.1 TCI-CD INTERFACE

RESULT	OPERATION	PARAMETERS	NOTES	§ [6]
Value	<i>decode</i>	(in TriMessageType message, in Type decodingHypothesis)	TCI-CD provided	7.3.2.2.1
TriMessageType	<i>encode</i>	(in Value value)		7.3.2.2.2
TInteger	<i>decodeValue</i>	(in TriMessageType message, in Type decodingHypothesis in TString encodingInfo, out Value decodedValue)		7.3.2.2.3
TriMessageType	<i>encodeValue</i>	(in Value value, in TString encodingInfo)		7.3.2.2.4
Type	<i>getTypeForName</i>	(in TString typeName)	TCI-CD required	7.3.2.1.1
Type	<i>getInteger</i>	()		7.3.2.1.2
Type	<i>getFloat</i>	()		7.3.2.1.3
Type	<i>getBoolean</i>	()		7.3.2.1.4
Type	<i>getCharstring</i>	()		7.3.2.1.6
Type	<i>getUniversalCharstring</i>	()		7.3.2.1.7
Type	<i>getHexstring</i>	()		7.3.2.1.8
Type	<i>getBitstring</i>	()		7.3.2.1.9
Type	<i>getOctetstring</i>	()		7.3.2.1.10
Type	<i>getVerdict</i>	()		7.3.2.1.11
void	<i>tciErrorReq</i>	(in TString message)		7.3.2.1.12

B.1 JAVA MAPPING (package org.etsi.ttcn.tci)

IDL (java interface)	METHODS	NOTES	§ [6]
public interface Type	public TciModuleId getDefiningModule (); public String getName (); public int getTypeClass (); public Value newInstance (); public String getTypeEncoding (); public String getTypeEncodingVariant(); public String[] getEncodeAttributes(); public String[] getVariantAttributes (String encoding); public String[] getTypeExtension(); public Value parseValue (String val); public MatchingMechanism newTemplate (int matchingType); public RangeBoundary getLowerTypeBoundary(); public RangeBoundary getUpperTypeBoundary(); public LengthRestriction getTypeLengthRestriction(); public MatchingMechanism getTypeMatchingMechanism();	NEW NEW NEW NEW NEW NEW	8.3.3.1
public interface Value	public Type getType (); public boolean notPresent (); public String getValueEncoding (); public String getValueEncodingVariant(); public String[] getEncodeAttributes(); public String[] getVariantAttributes (String encoding); public boolean isMatchingSymbol (); public String valueToString (); public boolean isLazy (); public boolean isFuzzy (); public boolean isEvaluated (); public LengthRestriction getLengthRestriction (); public LengthRestriction newLengthRestriction (); public void setLengthRestriction (LengthRestriction restriction); public boolean isIfPresentEnabled (); public void setIfPresentEnabled (boolean enabled); public boolean isOptional ();	NEW NEW NEW	8.3.4.1

public interface IntegerValue	public void setInteger(int value); public int getInteger();		8.3.4.2
public interface FloatValue	public void setFloat(float value); public float getFloat(); public void setDouble(double value); public float getDouble(); public void setBigDecimal(BigDecimal value); public BigDecimal getBigDecimal();		8.3.4.3
public interface BooleanValue	public void setBoolean(boolean value); public boolean getBoolean();		8.3.4.4
public interface CharstringValue	String getString (); void setString (String value); char getChar (int position); void setChar (int position, char value); int getLength (); void setLength (int len);		8.3.4.5
public interface BitstringValue	String getString (); void setString (String value); int getBit (int position); void setBit (int position, int value); int getLength (); void setLength (int len); boolean isMatchingAt (int position); MatchingMechanism getMatchingAt (int position); void setMatchingAt (int position, MatchingMechanism template);		8.3.4.6
public interface OctetstringValue	String getString (); void setString (String value); int getOctet (int position); void setOctet (int position, int value); int getLength (); void setLength (int len); boolean isMatchingAt (int position); MatchingMechanism getMatchingAt (int position); void setMatchingAt (int position, MatchingMechanism template);		8.3.4.7
public interface UniversalCharstringValue	String getString (); void setString (String value); int getChar (int position); void setChar (int position, int value); int getLength (); void setLength (int len);		8.3.4.8
public interface HexstringValue	String getString (); void setString (String value); int getHex (int position); void setHex (int position, int value); int getLength (); void setLength (int len); boolean isMatchingAt (int position); MatchingMechanism getMatchingAt (int position); void setMatchingAt (int position, MatchingMechanism template);		8.3.4.9
public interface RecordValue	public Value getField(String fieldName) ; public void setField(String fieldName, Value value) ; public String[] getFieldNames() ; public void setFieldOmitted(String fieldName);		8.3.4.10
public interface RecordOfValue	public Value getField(String fieldName) ; public void setField (int position, Value value) ; public void appendField (Value value) ; public Type getElementType() ; public int getLength() ; public void setLength(int len) ; public int getOffset() ; public int getPermutationCount () ; public Permutation getPermutation (int index); public Permutation newPermutation () ; public void definePermutation (Permutation permutation); public void removePermutation (int index); public void clearPermutations () ;		8.3.4.11
public interface UnionValue	Value getVariant (String variantName); void setVariant (String variantName, Value value); String getPresentVariantName (); String[] getVariantNames ();		8.3.4.12
public interface EnumeratedValue	String getEnum (); void setEnum (String enumValue); int getInt(); void setInt(int intValue);		8.3.4.13
public interface VerdictValue	public static final int NONE = 0; public static final int PASS = 1; public static final int INCONC = 2; public static final int FAIL = 3; public static final int ERROR = 4; public int getVerdict() ; public void setVerdict(int verdict) ;		8.3.4.14
public interface AddressValue	public int getAddress() ; public void setAddress(Value value) ;		8.3.4.15
TBoolean TFloat TInteger TString TStringSeq	boolean float int java.lang.String java.lang.String[]	Basic type mapping	8.3.1

A.2a triPlatform INTERFACE

RESULT	OPERATION	PARAMETERS	NOTES	§ [5]
TriStatusType	<i>triPReset</i>	()	Reset	5.6.1
TriStatusType	<i>triStartTimer</i>	(in TriTimerIdType <i>timerId</i> , in TriTimerDurationType <i>timerDuration</i>);	Timer handling	5.6.2.1
TriStatusType	<i>triStopTimer</i>	(in TriTimerIdType <i>timerId</i>);		5.6.2.2
TriStatusType	<i>triReadTimer</i>	(in TriTimerIdType <i>timerId</i> , out TriTimerDurationType <i>elapsedTime</i>);		5.6.2.3
TriStatusType	<i>triTimerRunning</i>	(in TriTimerIdType <i>timerId</i> , out boolean <i>running</i>);		5.6.2.4
void	<i>triTimeout</i>	(in TriTimerIdType <i>timerId</i>);		5.6.2.5
TriStatusType	<i>triExternalFunction</i>	(in TriFunctionIdType <i>functionId</i> , inout TriParameterListType <i>parameterList</i> , out TriParameterType <i>returnValue</i>);	Miscellaneous	5.6.3.1
TriComponentId	<i>triSelf</i>	()		5.6.3.2
TriMessage	<i>triRnd</i>	(in TriComponentIdType <i>componentId</i> , in TriMessage <i>seed</i>)		5.6.3.3
void	<i>triPErrorReq</i>	(in string <i>message</i>);	Error Handling	5.2.2

A.2b triCommunication INTERFACE

RESULT	OPERATION	PARAMETERS	NOTES	§ [5]
TriStatusType	<i>triSAReset</i>	()	TE -> SA (SAprovided)	5.5.1
TriStatusType	<i>triExecuteTestCase</i>	(in TriTestCaselIdType <i>testCaselId</i> , in TriPortIdListType <i>tsiPortList</i>);		5.5.2.1
TriStatusType	<i>triMap</i>	(in TriPortIdType <i>compPortId</i> , in TriPortIdType <i>tsiPortId</i>)		5.5.2.2
TriStatusType	<i>triMapParam</i>	(in TriPortIdType <i>compPortId</i> , in TriPortIdType <i>tsiPortId</i> , in TriParameterListType <i>paramList</i>);		5.5.2.3
TriStatusType	<i>triUnmap</i>	(in TriPortIdType <i>compPortId</i> , in TriPortIdType <i>tsiPortId</i>);		5.5.2.4
TriStatusType	<i>triUnmapParam</i>	(in TriPortIdType <i>compPortId</i> , in TriPortIdType <i>tsiPortId</i> , in TriParameterListType <i>paramList</i>);		5.5.2.5
TriStatusType	<i>triEndTestCase</i>	()		5.5.2.6
TriStatusType	<i>triSend</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriMessageType <i>sendMessage</i>);		5.5.3.1
TriStatusType	<i>triSendBC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriMessageType <i>sendMessage</i>);		5.5.3.2
TriStatusType	<i>triSendMC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressListType <i>SUAddresses</i> , in TriMessageType <i>sendMessage</i>);		5.5.3.3
TriStatusType	<i>triCall</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i>);		5.5.4.1
TriStatusType	<i>triCallBC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i>);		5.5.4.2
TriStatusType	<i>triCallMC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressListType <i>SUAddresses</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i>);		5.5.4.3
TriStatusType	<i>triReply</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i> , in TriParameterType <i>returnValue</i>);		5.5.4.4
TriStatusType	<i>triReplyBC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i> , in TriParameterType <i>returnValue</i>);		5.5.4.5
TriStatusType	<i>triReplyMC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressListType <i>SUAddresses</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i> , in TriParameterType <i>returnValue</i>);		5.5.4.6
TriStatusType	<i>triRaise</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriSignatureIdType <i>signatureId</i> , in TriExceptionType <i>exc</i>);		5.5.4.7
TriStatusType	<i>triRaiseBC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriSignatureIdType <i>signatureId</i> , in TriExceptionType <i>exc</i>);		5.5.4.8
TriStatusType	<i>triRaiseMC</i>	(in TriComponentIdType <i>componentId</i> , in TriPortIdType <i>tsiPortId</i> , in TriAddressListType <i>SUAddresses</i> , in TriSignatureIdType <i>signatureId</i> , in TriExceptionType <i>exc</i>);		5.5.4.9
TriStatusType	<i>triSUTactionInformal</i>	(in string <i>description</i>);		5.5.5.1
void	<i>triEnqueueMsg</i>	(in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriComponentIdType <i>componentId</i> , in TriMessageType <i>receivedMessage</i>);	SA -> TE (SArequired)	5.5.3.4
void	<i>triEnqueueCall</i>	(in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriComponentIdType <i>componentId</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i>);		5.5.4.10
void	<i>triEnqueueReply</i>	(in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriComponentIdType <i>componentId</i> , in TriSignatureIdType <i>signatureId</i> , in TriParameterListType <i>parameterList</i> , in TriParameterType <i>returnValue</i>);		5.5.4.11
void	<i>triEnqueueException</i>	(in TriPortIdType <i>tsiPortId</i> , in TriAddressType <i>SUAddress</i> , in TriComponentIdType <i>componentId</i> , in TriSignatureIdType <i>signatureId</i> , in TriExceptionType <i>exc</i>);		5.5.4.12
void	<i>triSAErrorReq</i>	(in string <i>message</i>);		5.2.1

B.2 JAVA MAPPING (package org.etsi.ttcn.tri)

IDL (java interface)	METHODS	NOTES	§ [5]
public interface TriBoolean	public void setBooleanValue (boolean value); public boolean getBooleanValue ();		6.3.1.1
string	java.lang.String	no range checking / bounds for characters in string	6.3.1.2
public interface TriPortId	public String getPortName (); public String getPortTypeName (); public TriComponentId getComponent (); public boolean isArray (); public int getPortIndex ();		6.3.2.1
public interface TriPortIdList	public int size (); public boolean isEmpty (); public java.util.Enumeration getPortIds (); public TriPortId get (int index);		6.3.2.2
public interface TriComponentId	public String getComponentId (); public String getComponentName (); public String getComponentTypeName (); public TriPortIdList getPortList (); public boolean equals (TriComponentId component);		6.3.2.3
public interface TriComponentIdListType	public int size (); public boolean isEmpty (); public java.util.Enumeration getComponents (); public TriComponentId get (int index); public void clear (); public void add (TriComponentId comp);		6.3.2.4
public interface TriMessage	public byte[] getEncodedMessage (); public void setEncodedMessage (byte[] message); public int getNumberOfBits (); public void setNumberOfBits (int amount); public boolean equals (TriMessage message);		6.3.2.5
public interface TriAddress	public byte[] getEncodedAddress (); public void setEncodedAddress (byte[] address); public int getNumberOfBits (); public void setNumberOfBits (int amount); public boolean equals (TriAddress address);		6.3.2.6
public interface TriAddressListType	public int size(); public boolean isEmpty (); public java.util.Enumeration getAddresses (); public TriAddress get (int index); public void clear (); public void add (TriAddress addr);		6.3.2.7
public interface TriSignatureId	public String getSignatureName (); public void setSignatureName (String sigName); public boolean equals (TriSignatureId sig);		6.3.2.8
public interface TriParameter	public String getParameterName (); public void setParameterName (String name); public int getNumberOfBits (); public void setNumberOfBits (int amount); public int getParameterPassingMode (); public void setParameterPassingMode (TriParameterPassingMode mode); public byte[] getEncodedParameter (); public void setEncodedParameter (byte[] parameter);		6.3.2.9
public interface TriParameterPassingMode	public final static int TRI_IN = 0; public final static int TRI_INOUT = 1; public final static int TRI_OUT = 2;		6.3.2.10
public interface TriParameterList	public int size (); public boolean isEmpty (); public java.util.Enumeration getParameters (); public TriParameter get (int index); public void clear (); public void add (TriParameter parameter);		6.3.2.11
public interface TriException	public byte[] getEncodedException (); public void setEncodedException (byte[] message); public int getNumberOfBits (); public void setNumberOfBits (int amount); public boolean equals (TriException exc);		6.3.2.12
public interface TriTimerId	public String getTimerName (); public boolean equals (TriTimerId timer);		6.3.2.13
public interface TriTimerDuration	public double getDuration(); public void setDuration(double duration); public boolean equals(TriTimerDuration duration);		6.3.2.14
public interface TriFunctionId	public String toString(); public String getFunctionName(); public boolean equals(TriFunctionId fun);		6.3.2.15
public interface TriTestCaseld	public String toString (); public String getTestCaseName (); public boolean equals (TriTestCaseld tc);		6.3.2.16
public interface TriStatus	public final static int TRI_OK = 0; public final static int TRI_ERROR = -1; public String toString (); public int getStatus(); public void setStatus (int status); public boolean equals (TriStatus status);		6.3.2.18

A.3 TCI-TL INTERFACE

OPERATION	NOTES	§ [6]
tliTExecute	testcases	7.3.4.1.1
tliTStart		7.3.4.1.2
tliTStop		7.3.4.1.3
tliTStarted		7.3.4.1.4
tliTTerminated		7.3.4.1.5
tliCtrlStart	control	7.3.4.1.6
tliCtrlStop		7.3.4.1.7
tliCtrlTerminated		7.3.4.1.8
tliMSend_m	asynchronous communication	7.3.4.1.9
tliMSend_m_BC		7.3.4.1.10
tliMSend_m_MC		7.3.4.1.11
tliMSend_c		7.3.4.1.12
tliMSend_c_BC		7.3.4.1.13
tliMSend_c_MC		7.3.4.1.14
tliMDetected_m		7.3.4.1.15
tliMDetected_c		7.3.4.1.16
tliMMismatch_m		7.3.4.1.17
tliMMismatch_c		7.3.4.1.18
tliMReceive_m		7.3.4.1.19
tliMReceive_c		7.3.4.1.20
tliPrCall_m	synchronous communication	7.3.4.1.21
tliPrCall_m_BC		7.3.4.1.22
tliPrCall_m_MC		7.3.4.1.23
tliPrCall_c		7.3.4.1.24
tliPrCall_c_BC		7.3.4.1.25
tliPrCall_c_MC		7.3.4.1.26
tliPrGetCallDetected_m		7.3.4.1.27
tliPrGetCallDetected_c		7.3.4.1.28
tliPrGetCallMismatch_m		7.3.4.1.29
tliPrGetCallMismatch_c		7.3.4.1.30
tliPrGetCall_m		7.3.4.1.31
tliPrGetCall_c		7.3.4.1.32
tliPrReply_m		7.3.4.1.33
tliPrReply_m_BC		7.3.4.1.34
tliPrReply_m_MC		7.3.4.1.35
tliPrReply_c		7.3.4.1.36
tliPrReply_c_BC		7.3.4.1.37
tliPrReply_c_MC		7.3.4.1.38
tliPrGetReplyDetected_m		7.3.4.1.39
tliPrGetReplyDetected_c		7.3.4.1.40
tliPrGetReplyMismatch_m		7.3.4.1.41
tliPrGetReplyMismatch_c		7.3.4.1.42
tliPrGetReply_m		7.3.4.1.43
tliPrGetReply_c		7.3.4.1.44
tliPrRaise_m		7.3.4.1.45
tliPrRaise_m_BC		7.3.4.1.46
tliPrRaise_m_MC		7.3.4.1.47
tliPrRaise_c		7.3.4.1.48
tliPrRaise_c_BC	7.3.4.1.49	
tliPrRaise_c_MC	7.3.4.1.50	
tliPrCatchDetected_m	7.3.4.1.51	
tliPrCatchDetected_c	7.3.4.1.52	
tliPrCatchMismatch_m	7.3.4.1.53	
tliPrCatchMismatch_c	7.3.4.1.54	
tliPrCatch_m	7.3.4.1.55	
tliPrCatch_c	7.3.4.1.56	
tliPrCatchTimeoutDetected	7.3.4.1.57	
tliPrCatchTimeout	7.3.4.1.58	

OPERATION	NOTES	§ [6]	
tliCreate	components	7.3.4.1.59	
tliCStart		7.3.4.1.60	
tliCRunning		7.3.4.1.61	
tliCAlive		7.3.4.1.62	
tliCStop		7.3.4.1.63	
tliCKill		7.3.4.1.64	
tliCDoneMismatch		7.3.4.1.65	
tliCDone		7.3.4.1.66	
tliCKilledMismatch		7.3.4.1.67	
tliCKilled		7.3.4.1.68	
tliCTerminated		7.3.4.1.69	
tliPConnect		ports	7.3.4.1.70
tliPDisconnect			7.3.4.1.71
tliPMap			7.3.4.1.72
tliPMapParam			7.3.4.1.73
tliPUnmap			7.3.4.1.74
tliPUnmapParam			7.3.4.1.75
tliPClear			7.3.4.1.76
tliPStart			7.3.4.1.77
tliPStop	7.3.4.1.78		
tliPHalt	7.3.4.1.79		
tliEncode	codec	7.3.4.1.80	
tliDecode		7.3.4.1.81	
tliTimeoutDetected	timers	7.3.4.1.82	
tliTimeoutMismatch		7.3.4.1.83	
tliTimeout		7.3.4.1.84	
tliTStart		7.3.4.1.85	
tliTStop		7.3.4.1.86	
tliTRead		7.3.4.1.87	
tliTRunning	7.3.4.1.88		
tliSEnter	scope	7.3.4.1.89	
tliSLeave		7.3.4.1.90	
tliVar	variables	7.3.4.1.91	
tliModulePar	module par	7.3.4.1.92	
tliGetVerdict	verdicts	7.3.4.1.93	
tliSetVerdict		7.3.4.1.94	
tliLog	log	7.3.4.1.95	
tliAEnter	alt	7.3.4.1.96	
tliALeave		7.3.4.1.97	
tliANomatch		7.3.4.1.98	
tliARepeat		7.3.4.1.99	
tliADefaults		7.3.4.1.100	
tliAActivate	7.3.4.1.101		
tliADeactivate	7.3.4.1.102		
tliAwait	7.3.4.1.103		
tliAction	action	7.3.4.1.104	
tliMatch	match	7.3.4.1.105	
tliMatchMismatch	operation	7.3.4.1.106	
tliInfo	tool dependent	7.3.4.1.107	
tliMChecked_m	check operation	7.3.4.1.108	
tliMChecked_c		7.3.4.1.109	
tliPrGetCallChecked_m		7.3.4.1.110	
tliPrGetCallChecked_c		7.3.4.1.111	
tliPrGetReplyChecked_m		7.3.4.1.112	
tliPrGetReplyChecked_c		7.3.4.1.113	
tliPrCatchChecked_m		7.3.4.1.114	
tliPrCatchChecked_c		7.3.4.1.115	
tliCheckedAny_m		7.3.4.1.116	
tliCheckedAny_c		7.3.4.1.117	
tliCheckAnyMismatch_m	7.3.4.1.118		
tliCheckAnyMismatch_c	7.3.4.1.119		
tliRnd		7.3.4.1.120	
tliEvaluate		7.3.4.1.121	

W3C XML mapping of the operations on the logging interface is provided in § [11.4 \[6\]](#).

B.3 TCI-TL TYPES (W3C XML mapping)

IDL simple types	XSD	XSD string enumeration values	NOTES	§ [6]
<i>TBoolean</i>	boolean			11.3.1.1
<i>TString</i>	string			11.3.1.2
<i>TInteger</i>	integer			11.3.1.3
<i>TriTimerDuration</i>	float			11.3.1.4
<i>TciParameterPassingModeType</i>	string	"in", "out", "inout"		11.3.1.5
<i>TriStatusType</i>	string	"TRI_OK", "TRI_Error"		11.3.1.6
<i>TciStatusType</i>	string	"TCI_OK", "TCI_Error"		11.3.1.7
<i>ComponentStatusType</i>	string	"inactiveC", "running", "stoppedC", "killedC", "nullC"		11.3.1.8
<i>TimerStatusType</i>	string	"runningT", "inactiveT", "expiredT", "nullT"		11.3.1.9
<i>PortStatusType</i>	string	"startedP", "haltedP", "stopped"		11.3.1.10
IDL complex types	XSD sequence elements	XSD attributes	NOTES	§ [6]
<i>TriPortIdType</i>	comp, port			11.3.2.1
<i>TriComponentIdType</i>	id, null			11.3.2.2
<i>TriComponentIdListType</i>	comp			11.3.2.3
<i>Port</i>	id, index			11.3.2.4
<i>Id</i>	name, id, type			11.3.2.5
<i>TriMessageType</i>		val, paddingBits		11.3.2.6
<i>TriParameterType</i>		val, paddingBits, name, mode		11.3.2.7
<i>TriParameterListType</i>	par			11.3.2.8
<i>TriAddressType</i>		val, paddingBits		11.3.2.9
<i>TriAddressListType</i>	addr			11.3.2.10
<i>TriExceptionType</i>		val, paddingBits		11.3.2.11
<i>TriSignatureIdType</i>		val		11.3.2.12
<i>TriTimerIdType</i>	id			11.3.2.13
<i>TriTimerDurationType</i>			restriction float	11.3.2.14
<i>QualifiedName</i>		moduleName, baseName		11.3.2.15
<i>TciBehaviourType</i>	name			11.3.2.16
<i>TciTestCaseldType</i>	name			11.3.2.17
<i>TciParameterType</i>	val	name, mode		11.3.2.18
<i>TciParameterListType</i>	par			11.3.2.19
<i>TriPortIdListType</i>	port			11.3.2.20
IDL abstract values	XSD choice elements	XSD attributes	NOTES	§ [6]
<i>Value</i>	integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_charstring, record, record_of, array, set, set_of, enumerated, union, anytype, address, component, port, default, timer			11.3.3.1
<i>IntegerValue</i>				11.3.3.2
<i>FloatValue</i>				11.3.3.3
<i>BooleanValue</i>				11.3.3.4
<i>VerdictValue</i>				11.3.3.6
<i>BitstringValue</i>				11.3.3.7
<i>HexstringValue</i>				11.3.3.8
<i>OctetstringValue</i>				11.3.3.9
<i>CharstringValue</i>				11.3.3.10
<i>UniversalCharstringValue</i>				11.3.3.11
<i>RecordValue</i>	integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_charstring, record, record_of, array, set, set_of, enumerated, union, anytype, address, component, default, null, omit, matching_symbol, not_evaluated	name, type, module, modifier, annotation		11.3.3.12
<i>RecordOfValue</i>				11.3.3.13
<i>SetValue</i>				11.3.3.15
<i>SetOfValue</i>				11.3.3.16
<i>UnionValue</i>				11.3.3.18
<i>ArrayValue</i>	integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_charstring, record, record_of, array, set, set_of, enumerated, union, anytype, address, component, port, default, timer, null, omit, matching_symbol, not_evaluated			11.3.3.14
<i>EnumeratedValue</i>	value, intValue, null, omit, matching_symbol, not_evaluated			11.3.3.17
<i>AnytypeValue</i>	integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_charstring, record, record_of, array, set, set_of, enumerated, union, address, null, omit, matching_symbol, not_evaluated			11.3.3.19
<i>AddressValue</i>	integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_charstring, record, record_of, array, set, set_of, enumerated, union, anytype, null, omit, matching_symbol, not_evaluated			11.3.3.20
<i>ComponentValue</i>	value, null, omit, matching_symbol, not_evaluated			11.3.3.21
<i>PortValue</i>				11.3.3.22
<i>TimerValue</i>	value, null, omit			11.3.3.24
<i>DefaultValue</i>	value, null, omit, not_evaluated			11.3.3.23
IDL abstract logging type	XSD elements	XSD attributes	NOTES	§ [6]
<i>TciValueTemplate</i>	choice: integer, float, boolean, verdicttype, bitstring, hexstring, octetstring, charstring, universal_string, record, record_of, array, set, set_of, enumerated, union, anytype, address, omit, any, anyoromit, templateDef			11.3.4.1
<i>TciNonValueTemplate</i>	choice: any, all, templateDef, null			11.3.4.2
<i>TciValueList</i>	sequence: val			11.3.4.3
<i>TciValueDifference</i>	sequence: val, tmpl	name, type, module, annotation, desc		11.3.4.4
<i>TciValueDifferenceList</i>	sequence: diff			11.3.4.5

Related EVENTS



10 October 2017 (STV)

5th **UCAAT** User Conference on
Advanced Automated Testing

ETSI

Berlin, 11-13 October 2017

Hosted by: **Fraunhofer**
FOKUS

11-13 October 2017 (UCAAT)

NOTES:

This Reference Card summarizes interfaces, operations and language mappings to support developers of TTCN-3 Codec and adaptations. The document is not part of a standard, not warranted error-free, and a 'work in progress'. For comments or suggestions, please contact the editors via ttn3-qrc@blukactus.com.

Numbers in the right-hand column of the tables refer to sections or annex in ETSI standards ES 201873-5 and -6.

NEW Interface operations/types introduced/modified in edition 4.9.1 have been marked.