ETSI ES 201 873-6 V4.8.1 (2016-07)

Methods for Testing and Specification (MTS);

The Testing and Test Control Notation version 3;

Part 6: TTCN‑3 Control Interface (TCI)

**ETSI Standard**

Reference

RES/MTS-201873-6 T3ed481TCI

Keywords

control, interface, methodology, TCI, 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.

© European Telecommunications Standards Institute 2016.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
**3GPP**TM and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and
of the 3GPP Organizational Partners.
**GSM**® and the GSM logo are Trade Marks registered and owned by the GSM Association.

# 9 ANSI C language mapping

## 9.1 Introduction

This clause defines the TCI ANSI‑C [8] language mapping for the TCI data specified in clause 7.2 and for the TCI operations specified in clause 7.3.

## 9.2 Value interfaces

Table

| TCI IDL Interface | ANSI C representation | Notes and comments |
| --- | --- | --- |
| Type |
| TciModuleIdType getDefiningModule() | TciModuleIdTypetciGetDefiningModule(Type inst) |  |
| Tstring getName() | String tciGetName(Type inst) | String type reused from IDL (OMG recommendation). |
| TciTypeClassType getTypeClass() | TciTypeClassType tciGetTypeClass (Type inst) |  |
| Value newInstance() | Value tciNewInstance(Type inst) |  |
| Tstring getTypeEncoding() | String tciGetTypeEncoding(Type inst) |  |
| TstringSeq getTypeExtension() | String\* tciGetTypeExtension(Type inst) |  |
| Tstring getTypeEncodingVariant() | String tciGetTypeEncodingVariant(Type inst) |  |
| Value parseValue(TString val) | Value tciParseValue(Type inst, String val) |  |
| LengthRestriction getTypeLengthRestriction() | int tciGetTypeLengthRestriction(Type inst, TciLengthRestriction \* restriction) | Returns 0 for no restriction and -1 for restriction. The restriction is returned in the second parameter. |
| RangeBoundary getLowerTypeBoundary() | Int tciGetLowerTypeBoundary(Type inst, TciRangeBoundary \* boundary) | Returns 0 for no boundary and -1 if boundary is present. The boundary is returned in the second parameter. |
| RangeBoundary geUpperTypeBoundary() | int tciGetUpperTypeBoundary(Type inst, TciRangeBoundary \* boundary) |
| MatchingMechanism getTypeMatchingMechanism() | Value tciGetTypeMatchingMechanism(Type inst) | Get the restriction of type of value |
| Value |
| Tstring getValueEncoding() | String tciGetValueEncoding(Value inst) |  |
| Tstring getValueEncodingVariant() | String tciGetValueEncodingVariant(Value inst) |  |
| Type getType() | Type tciGetType(Value inst) |  |
| Tboolean notPresent() | Boolean tciNotPresent(Value inst) | Boolean type reused from IDL (OMG recommendation). |
|  | void tciSetNull(Value inst) | For optional parameters of operations, see clause 9.7. |
|  | Boolean tciIsNull(Value inst) | For optional parameters of operations, see clause 9.7. Boolean type reused from IDL (OMG recommendation). |
| Tboolean isMatchingSymbol() | Boolean tciIsMatchingSymbol(Value inst) |  |
| TString valueToString() | String tciValueToString(Value inst) |  |
| Tboolean isLazy () | Boolean tciIsLazy(Value inst) |  |
| Tboolean isFuzzy () | Boolean tciIsFuzzy(Value inst) |  |
| Tboolean isEvaluated() | Boolean tciIsEvaluated(Value inst) |  |
|  | Boolean tciHasLengthRestriction(Value inst) | Returns true if length restriction is present, false otherwise. |
| LengthRestriction getLengthRestriction() | TciLengthRestriction tciGetLengthRestriction(Value inst) |  |
|  | void tciRemoveLengthRestriction(Value inst) | Removes length restriction. |
| void setLengthRestriction (LengthRestriction restriction) | void tciSetLengthRestriction(Value inst, TciLengthRestriction \* restriction) |  |
| TBoolean isIfPresentEnabled() | Boolean tciIsIfPresentEnabled(Value inst) |  |
| void setIfPresentEnabled(TBoolean enabled) | void tciSetIfPresentEnabled(Value inst, Boolean enabled) |  |
| LengthRestriction getTypeLengthRestriction() | int tciGetValueLengthRestriction(Value inst, TciLengthRestriction \* restriction) | Returns 0 for no restriction and -1 for restriction. The restriction is returned in the second parameter. |
| RangeBoundary getLowerTypeBoundary() | int tciGetLowerValueBoundary(Value inst, TciRangeBoundary \* boundary) | Returns 0 for no boundary and -1 if boundary is present. The boundary is returned in the second parameter. |
| RangeBoundary geUpperTypeBoundary() | int tciGetUpperValueBoundary(Value inst, TciRangeBoundary \* boundary) |
| MatchingMechanism getTypeMatchingMechanism() | Value tciGetValueMatchingMechanism(Value inst) | Get the restriction of type of value |
| IntegerValue |
| Tinteger getInt() | String tciGetIntAbs(Value inst) | Returns the (10‑base) integer absolute value as an ASCII string. |
|  | unsigned long int tciGetIntNumberOfDigits (Value inst) | Returns the number of digits in an integer value. |
|  | Boolean tciGetIntSign(Value inst) | Returns true if the number is positive, false otherwise. |
|  | char tciGetIntDigit (Value inst,  unsigned long int position) | Returns the value of the digit at position 'position', where position 0 is the least significant digit. |
|  | long long tciGetInt(Value inst) | Alternative getInt realization for integers not exceeding the boundaries of signed 64-bit. |
| void setInt(in Tinteger value) | void tciSetIntAbs(Value inst, String value) | Sets the (10‑base) absolute value of the integer via an ASCII string. The first digit shall not be 0 unless the value is 0. |
|  | Void tciSetIntNumberOfDigits (Value inst,  unsigned long int dig\_num) | Sets the number of digits in an integer value. |
|  | void tciSetIntSign (Value inst,  Boolean sign) | Sets the sign to + (true) or - (false). |
|  | void tciSetIntDigit (Value inst,  unsigned long int position,  char digit) | Sets the value of the digit at position 'position', where position 0 is the least significant digit. |
|  | void tciSetInt(Value inst, long long value) | Alternative setInt realization for integers not exceeding the boundaries of signed 64-bit. |
| FloatValue |
| Tfloat getFloat() | double tciGetFloatValue(Value inst) |  |
| void setFloat(in Tfloat value) | void tciSetFloatValue(Value inst, double value) |  |
| BooleanValue |
| Tboolean getBoolean() | Boolean tciGetBooleanValue(Value inst) |  |
| void setBoolean (in Tboolean value) | void tciSetBooleanValue (Value inst, Boolean value) |  |
| CharstringValue |
| Tstring getString() | TciCharStringValue tciGetCStringValue(Value inst) |  |
| void setString(in Tstring value) | void tciSetCStringValue (Value inst, TciCharStringValue value) |  |
| Tchar getChar (in Tinteger position) | char tciGetCStringCharValue (Value inst, long int position) |  |
| void setChar(in Tinteger position, in Tchar value) | void tciSetCStringCharValue (Value inst,long int position,char value) |  |
| Tinteger getLength() | unsigned long int tciGetCStringLength (Value inst) |  |
| void setLength(in Tinteger len) | void tciSetCStringLength (Value inst,unsigned long int len) |  |
| UniversalCharstringValue |
| Tstring getString() | TciUCStringValue tciGetUCStringValue(Value inst) |  |
| void setString(in Tstring value) | void tciSetUCStringValue (Value inst, TciUCStringValue value) |  |
| TuniversalChar getChar(in Tinteger position) | void tciGetUCStringCharValue (Value inst, unsigned long int position,  TciUCValue result) |  |
| void setChar(in Tinteger position, in TuniversalChar value) | void tciSetUCStringCharValue (Value inst,unsigned long int position, TciUCValue value) |  |
| Tinteger getLength() | unsigned long int tciGetUCStringLength(Value inst) |  |
| void setLength(in Tinteger len) | void tciSetUCStringLength (Value inst,unsigned long int len) |  |
| BitstringValue |
| Tstring getString() | String tciGetBStringValue(Value inst) |  |
| void setString(in Tstring value) | void tciSetBStringValue (Value inst, String value) |  |
| Tchar getBit (in integer position) | int tciGetBStringBitValue (Value inst, long int position) |  |
| void setBit(in Tinteger position,in Tinteger value) | void tciSetBStringBitValue (Value inst,unsigned long int position,int value) |  |
| Tinteger getLength() | unsigned long int tciGetBStringLength(Value inst) |  |
| void setLength(in Tinteger len) | void tciSetBStringLength (Value inst,long int len) |  |
| TBoolean isMatchingAt(in TInteger position) | Boolean tciIsBStringMatchingAt(Value inst, unsigned long int position) |  |
| MatchingMechanism getMatchingAt(in TInteger position) | Value tciGetBStringMatchingAt(Value inst, unsigned long int position) |  |
| void setMatching(in TInteger position, in MatchingMechanism template) | void tciSetBStringMatchingAt(Value inst, unsigned long int position, Value template) |  |
| HexstringValue |
| Tstring getString() | String tciGetHStringValue(Value inst) |  |
| void setString(in Tstring value) | void tciSetHStringValue (Value inst, String value) |  |
| Tchar getHex (in Tinteger position) | int tciGetHStringHexValue (Value inst, unsigned long int position) |  |
| void setBit(in Tinteger position,in Tinteger value) | void tciSetHStringHexValue (Value inst,unsigned long int position,int value) |  |
| Tinteger getLength() | long int tciGetHStringLength(Value inst) |  |
| void setLength(in Tinteger len) | void tciSetHStringLength (Value inst,unsigned long int len) |  |
| TBoolean isMatchingAt(in TInteger position) | Boolean tciIsHStringMatchingAt(Value inst, unsigned long int position) |  |
| MatchingMechanism getMatchingAt(in TInteger position) | Value tciGetHStringMatchingAt(Value inst, unsigned long int position) |  |
| void setMatching(in TInteger position, in MatchingMechanism template) | void tciSetHStringMatchingAt(Value inst, unsigned long int position, Value template) |  |
| OctetstringValue |
| Tstring getString() | String tciGetOStringValue(Value inst) |  |
| void setString(in Tstring value) | void tciSetOStringValue (Value inst, String value) |  |
| Tchar getOctet(in Tinteger position) | int tciGetOStringOctetValue (Value inst, unsigned long int position) |  |
| void setOctet(in Tinteger position,in Tinteger value) | void tciSetOStringOctetValue (Value inst,unsigned long int position,int value) |  |
| Tinteger getLength() | unsigned long int tciGetOStringLength(Value inst) |  |
| void setLength(in Tinteger len) | void tciSetOStringLength (Value inst,unsigned long int len) |  |
| TBoolean isMatchingAt(in TInteger position) | Boolean tciIsOStringMatchingAt(Value inst, unsigned long int position) |  |
| MatchingMechanism getMatchingAt(in TInteger position) | Value tciGetOStringMatchingAt(Value inst, unsigned long int position) |  |
| void setMatching(in TInteger position, in MatchingMechanism template) | void tciSetOStringMatchingAt(Value inst, unsigned long int position, Value template) |  |
| RecordValue |
| Value getField(in Tstring fieldName int) | Value tciGetRecFieldValue (Value inst,String fieldName int) |  |
| void setField(in Tstring fieldName int,in Value value) | void tciSetRecFieldValue (Value inst,String fieldName int,Value value) |  |
| Tstring[] getFieldNames() | char\*\* tciGetRecFieldNames(Value inst) | Returns a NULL‑terminated array of the field names. |
| void setFieldOmitted(in Tstring fieldName int) | void setFieldOmitted (Value inst,String fieldName int) |  |
| RecordOfValue |
| Value getField(in Tinteger position) | Value tciGetRecOfFieldValue (Value inst,unsigned long int position) |  |
| void setField(in Tinteger position,in Value value) | void tciSetRecOfFieldValue (Value inst,unsigned long int position,Value value) |  |
| void appendField(in Value value) | void tciAppendRecOfFieldValue (Value inst,Value value) |  |
| Type getElementType() | Type tciGetRecOfElementType(Value inst) |  |
| Tinteger getLength() | unsigned long int tciGetRecOfLength(Value inst) |  |
| void setLength(in Tinteger len) | void tciSetRecOfLength (Value inst,unsigned long int len) |  |
| Tinteger getOffset() | unsigned long int tciGetOffset(Value inst) |  |
| TInteger getPermutationCount() | unsigned long int tciGetPermutationCount(Value inst) |  |
| Permutation getPermutation(TInteger index) | TciPermutation tciGetPermutation(Value inst, unsigned long int index) |  |
| void definePermutation(Permutation permutation) | void tciDefinePermutation(Value inst, TciPermutation permutation) |  |
| void removePermutation(TInteger index) | void tciRemovePermutation(Value intst, unsigned long int index) |  |
| void clearPermutations() | void tciClearPermutations(Value inst) |  |
| UnionValue |
| Value getVariant(in Tstring variantName) | Value tciGetUnionVariant (Value inst,String variantName) |  |
| void setVariant(in Tstring variantName,in Value value) | void tciSetUnionVariant (Value inst,String variantName,Value value) |  |
| Tstring getPresentVariantName() | String tciGetUnionPresentVariantName (Value inst) |  |
| Tstring[] getVariantNames() | char\*\* tciGetUnionVariantNames(Value inst) | Returns a NULL‑terminated array of the field names. |
| EnumeratedValue |
| Tstring getEnum() | String tciGetEnumValue(Value inst) |  |
| void setEnum(in Tstring enumValue) | void tciSetEnumValue (Value inst,String enumValue) |  |
| Tinteger getInt() | unsigned long tciGetEnumInt(Value inst); |  |
| setInt(in Tinteger intValue) | void tciSetEnumInt(Value inst, unsigned long intValue); |  |
| VerdictValue |
| Tinteger getVerdict() | int tciGetVerdictValue(Value inst) |  |
| void setVerdict(in Tinteger verdict) | void tciSetVerdictValue(Value inst, int verdict) |  |
| AddressValue |
| Value getAddress() | Value tciGetAddressValue(Value inst) |  |
| void setAddress(in Value value) | void tciSetAddressValue(Value inst, Value value) |  |
| **MatchingMechanism** |
| TciMatchingTypeType getMatchingType() | TciMatchingTypeType tciGetMatchingType(Value inst) |  |
| **MatchingList** |
| TInteger size() | unsigned long int tciGetMatchingListSize(Value inst) |  |
| Value get(TInteger position) | Value tciGetMatchingListItem(Value inst, unsigned long int position) |  |
| void add(Value item) | void tciAddMatchingListItem(Value inst, Value item) |  |
| void remove(TInteger position) | void tciRemoveMatchingListItem(Value inst, unsigned long int position) |  |
| void clear() | void tciClearMatchingList(Value inst) |  |
| **ValueRange** |
| RangeBoundary getLowerBoundary() | TciRangeBoundary tciGetLowerRangeBoundary(Value inst) |  |
| RangeBoundary getUpperBoundary() | TciRangeBoundary tciGetUpperRangeBoundary(Value inst) |  |
| void setLowerBoundary (RangeBoundary lowerBoundary) | void setLowerRangeBoundary(Value inst, TciRangeBoundary lowerBoundary) |  |
| void setUpperBoundary (RangeBoundary upperBoundary) | void setUpperRangeBoundary(Value inst, TciRangeBoundary upperBoundary) |  |
| **CharacterPattern** |
| Value getPatternString () | Value tciGetPatternString(Value inst) |  |
| void setPatternString(Value characterPattern) | void tciSetPatternString(Value inst, Value characterPattern) |  |
| **MatchDecodedContent** |
| Value getContent() | Value tciGetDecodedMatchContent(Value inst) |  |
| void setContent(Value content) | void tciSetDecodedMatchContent(Value inst, Value content) |  |