ETSI ES 201 873-6 V4.10.1 (2018-05)

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 T3ed4A1

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.

© ETSI 2018.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.
**3GPP**TM and **LTE**TM are trademarks of ETSI registered for the benefit of its Members and
of the 3GPP Organizational Partners.
**oneM2M** logo is protected for the benefit of its Members.
**GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

#### 8.3.4.6 BitstringValue

**BitstringValue** is mapped to the following interface:

// TCI IDL BitstringValue

package org.etsi.ttcn.tci;

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);

 public java.io.InputStream getInputStream();

 public void setInputStream(java.io.InputStream stream, int numberOfBits);

 boolean isMatchingAt (int position);

 MatchingMechanism getMatchingAt (int position);

 void setMatchingAt (int position, MatchingMechanism template);

}

**Methods:**

* getString Returns the textual representation of this BitstringValue, as defined in TTCN‑3. E.g. the textual representation of 0101 is '0101'B. The textual representation of the empty TTCN‑3 bitstring is ''B, while its length is zero. In some cases, the message data are available in the form of a stream and cannot be converted into a string by TCi (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.
* setString Sets the value of this BitstringValue according to the textual representation as defined by value. E.g. The value of this BitstringValue will be 0101 if the textual representation in value is '0101'B.
* getBit Returns the value (0 | 1) at position of this TTCN‑3 bitstring. position 0 denotes the first bit of the TTCN‑3 bitstring. Valid values for position are 0 to length - 1.
* setBit Set the bit at position to value (0 | 1). position 0 denotes the first bit in this BitstringValue. Valid values for position are 0 to length - 1.
* getLength Returns the length of this BitstringValue in bits, zero if the value of this BitstringValue is omit.
* setLength Sets the length of this BitstringValue in bits to len.
* getInputStream Returns the bits in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the bitstring contains matching symbols.
* setInputStream Sets the value of this BitstringValue by providing a source stream that is used by the BitstringValue object to read the value content. Values set this way don’t contain matching symbols.
* isMatchingAt Returns true if the item at position of this TTCN‑3 bitstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* getMatchingAt If the position of this TTCN‑3 bitstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* setMatching Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.

#### 8.3.4.7 OctetstringValue

**OctetstringValue** is mapped to the following interface:

// TCI IDL OctetstringValue

package org.etsi.ttcn.tci;

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);

 public java.io.InputStream getInputStream();

 public void setInputStream(java.io.InputStream stream);

 boolean isMatchingAt (int position);

 MatchingMechanism getMatchingAt (int position);

 void setMatchingAt (int position, MatchingMechanism template);

}

**Methods:**

* getString Returns the textual representation of this OctetstringValue, as defined in TTCN‑3. E.g. the textual representation of 0xCAFFEE is 'CAFFEE'O. The textual representation of the empty TTCN‑3 octetstring is ''O, while its length is zero. In some cases, the message data are available in the form of a byte stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.
* setString Sets the value of this OctetstringValue according to the textual representation as defined by value. E.g. the value of this OctetstringValue will be 0xCAFFEE if the textual representation in value is 'CAFFEE'O.
* getOctet Returns the value (0..255) at position of this TTCN‑3 octetstring. position 0 denotes the first octet of the TTCN‑3 octetstring. Valid values for position are 0 to length - 1.
* setOctet Set the octet at position to value (0..255). position 0 denotes the first octet in the octetstring. Valid values for position are 0 to length - 1.
* getLength Returns the length of this OctetstringValue in octets, zero if the value of this OctetstringValue is omit.
* setLength Sets the length of this OctetstringValue in octets to len.
* getInputStream Returns the octets in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the octetstring contains matching symbols.
* setInputStream Sets the value of this OctetstringValue by providing a source stream that is used by the OctetstringValue object to read the value content. Values set this way don’t contain matching symbols.
* isMatchingAt Returns true if the item at position of this TTCN‑3 octetstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* getMatchingAt If the position of this TTCN‑3 octetstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* setMatching Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.

#### 8.3.4.9 HexstringValue

**HexstringValue** is mapped to the following interface:

// TCI IDL HexstringValue

package org.etsi.ttcn.tci;

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);

 public java.io.InputStream getInputStream();

 public void setInputStream(java.io.InputStream stream, int numberOfItems);

 boolean isMatchingAt (int position);

 MatchingMechanism getMatchingAt (int position);

 void setMatchingAt (int position, MatchingMechanism template);

}

**Methods:**

* getString Returns the textual representation of this HextstringValue, as defined in TTCN‑3. E.g. the textual representation of 0xAFFEE is 'AFFEE'H. The textual representation of the empty TTCN‑3 hexstring is ''H, while its length is zero. In some cases, the message data are available in the form of a byte stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.
* setString Sets the value of this HexstringValue according to the textual representation as defined by value. E.g. the value of this HexstringValue will be 0xAFFEE if the textual representation in value is 'AFFEE'H.
* getHex Returns the value (0...15) at position of this TTCN‑3 hexstring. position 0 denotes the first hex digits of the TTCN‑3 hexstring. Valid values for position are 0 to length - 1.
* setHex Set the hex digit at position to value (0...16). position 0 denotes the first octet in the hexstring. Valid values for position are 0 to length - 1.
* getLength Returns the length of this HexstringValue in octets, zero if the value of this HexstringValue is omit.
* setLength Sets the length of this HexstringValue in hex digits to len.
* getInputStream Returns the content in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the hexstring contains matching symbols.
* setInputStream Sets the value of this HexstringValue by providing a source stream that is used by the HexstringValue object to read the value content. Values set this way don’t contain matching symbols.
* isMatchingAt Returns true if the item at position of this TTCN‑3 hexstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* getMatchingAt If the position of this TTCN‑3 hexstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* setMatching Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.

#### 10.5.3.8 BitstringValue

TTCN-3 bitstring value support. It is mapped to the following pure virtual class:

class BitstringValue : public virtual TciValue {

public:

 virtual ~BitstringValue ();

 virtual Tbit getBit (Tindex p\_position) const =0;

 virtual Tsize getLength () const =0;

 virtual const Tstring & getString () const =0;

 virtual void setBit (Tindex p\_position, Tbit p\_bsValue)=0;

 virtual void setLength (Tindex p\_new\_length)=0;

 virtual void setString (const Tstring &p\_bsValue)=0;

 virtual std:istream \* getInputStream()=0;

 virtual void setInputStream(std:istream \* stream, Tsize bitLen)=0;

 virtual Tboolean isMatchingAt (Tindex p\_position) const =0;

 virtual MatchingMechanism & getMatchingAt (Tindex p\_position) const =0;

 virtual void setMatchingAt (Tindex p\_position, MatchingMechanism &p\_template) = 0;

 virtual Tboolean operator== (const BitstringValue &p\_bitStr) const =0;

 virtual BitstringValue \* clone () const =0;

 virtual Tboolean operator< (const BitstringValue &p\_bitStr) const =0;

}

**Methods:**

~BitstringValue

 Destructor

getBit

 Returns the bit at the specified position

getLength

 Returns the length of the string

getString

 Returns the value of the string. In some cases, the message data are available in the form of a stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.

setBit

 Sets the bit value at the specified position

setLength

 Sets the length of the string

setString

 Sets the string value

getInputStream
Returns the bits in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the bitstring contains matching symbols.

setInputStream
Sets the value of this BitstringValue by providing a source stream that is used by the BitstringValue object to read the value content. Values set this way don’t contain matching symbols.

isMatchingAt

 Returns true if the item at the specified position is a matching mechanism inside a value

getMatchingAt

 Returns a matching mechanism at the specified position

setMatchingAt

 Sets the matching mechanism at the specified position

operator==

 Returns true if both objects are equal

clone

 Returns a copy of the BitstringValue

operator<

 Operator < overload

#### 10.5.3.9 OctetstringValue

TTCN-3 octetstring value support. It is mapped to the following pure virtual class:

class OctetstringValue : public virtual TciValue {

public:

 virtual ~OctetstringValue ();

 virtual Tsize getLength () const =0;

 virtual const Tchar getOctet (Tindex p\_position) const =0;

 virtual const Tstring & getString () const =0;

 virtual void setLength (Tsize p\_length)=0;

 virtual void setOctet (Tindex p\_position, Tchar p\_ochar)=0;

 virtual void setString (const Tstring &p\_osValue)=0;

 virtual std:istream \* getInputStream()=0;

 virtual void setInputStream(std:istream \* stream)=0;

 virtual Tboolean isMatchingAt (Tindex p\_position) const =0;

 virtual MatchingMechanism & getMatchingAt (Tindex p\_position) const =0;

 virtual void setMatchingAt (Tindex p\_position, MatchingMechanism &p\_template) = 0;

 virtual Tboolean operator== (const OctetstringValue & p\_octStr) const =0;

 virtual OctetstringValue \* clone () const =0;

 virtual Tboolean operator< (const OctetstringValue & p\_octStr) const =0;

}

**Methods:**

~OctetstringValue

 Destructor

getLength

 Returns the length of the string

getOctet

 Returns the textual representation of the octetchar at the specified position

getString

 Returns the string value. In some cases, the message data are available in the form of a byte stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.

setLength

 Sets the length of the string

setOctet

 Sets the char at specified position

setString

 Sets the value of the string

getInputStream
Returns the octets in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the octetstring contains matching symbols.

setInputStream
Sets the value of this OctetstringValue by providing a source stream that is used by the OctetstringValue object to read the value content. Values set this way don’t contain matching symbols.

isMatchingAt

 Returns true if the item at the specified position is a matching mechanism inside a value

getMatchingAt

 Returns a matching mechanism at the specified position

setMatchingAt

 Sets the matching mechanism at the specified position

operator==

 Returns true if both objects are equal

clone

 Returns a copy of the OctetstringValue

operator<

 Operator < overload

#### 10.5.3.10 HexstringValue

TTCN-3 hexstring value support. It is mapped to the following pure virtual class:

class HexstringValue : public virtual TciValue {

public:

 virtual ~HexstringValue ();

 virtual Tchar getHex (Tindex p\_position) const =0;

 virtual Tsize getLength () const =0;

 virtual const Tstring & getString () const =0;

 virtual void setHex (Tindex p\_position, Tchar p\_hcValue)=0;

 virtual void setLength (Tsize p\_length)=0;

 virtual void setString (const Tstring &p\_hsValue)=0;

 virtual std:istream \* getInputStream()=0;

 virtual void setInputStream(std:istream \* stream, Tsize numberOfItems)=0;

 virtual Tboolean isMatchingAt (Tindex p\_position) const =0;

 virtual MatchingMechanism & getMatchingAt (Tindex p\_position) const =0;

 virtual void setMatchingAt (Tindex p\_position, MatchingMechanism &p\_template) = 0;

 virtual Tboolean operator== (const HexstringValue & p\_hexStr) const =0;

 virtual HexstringValue \* clone () const =0;

 virtual Tboolean operator< (const HexstringValue & p\_hexStr) const =0;

}

**Methods:**

~HexstringValue

 Destructor

getHex

 Returns the element at the specified position

getLength

 Returns the length of the string

getString

 Returns the string value. In some cases, the message data are available in the form of a stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this method returns null and the getInputStream method shall be used for reading the data.

setHex

 Sets the hex value at the specified position

setLength

 Sets the length of the string

setString

 Sets the value of the string

getInputStream
Returns the data in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the hexstring contains matching symbols.

setInputStream
Sets the value of this HexstringValue by providing a source stream that is used by the HexstringValue object to read the value content. Values set this way don’t contain matching symbols.

isMatchingAt

 Returns true if the item at the specified position is a matching mechanism inside a value

getMatchingAt

 Returns a matching mechanism at the specified position

setMatchingAt

 Sets the matching mechanism at the specified position

operator==

 Returns true if both objects are equal

clone

 Returns a copy of the HexstringValue

operator<

 Operator < overload

#### 12.4.4.6 BitstringValue

**BitstringValue** is mapped to the following interface:

public interface ITciBitstringValue : ITciValue {
 string StringValue { get; set; }
 byte this[int position] { get; set; }
 int Length { get; set; }
 bool IsMatchingAt (int position);

 System.IO.Stream GetInputStream ();

 void SetInputStream(System.IO.Stream stream, int numberOfBits);
 ITciMatchingMechanism GetMatchingAt (int position);

 void SetMatchingAt (int position, ITciMatchingMechanism template);

}

**Members:**

* StringValue
Gets or sets the string value of the TTCN-3 bitstring. The only allowed characters in the string passed to this property are '0' and '1'. The string returned by the property contains a sequence of '0' and '1' digits. In some cases, the message data are available in the form of a stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this property returns null and the GetInputStream method shall be used for reading the data.
* Indexing operator
Get or sets the value of the bit at the specified position. All non-zero values shall be interpreted as if the bit was present. IndexOutOfRangeException is thrown if the position is less than zero or greater or equal to the string length.
* Length
Gets or sets the length of this ITciBitstringValue in bits. The property returns zero if the value of this object is omit. In case the new length is greater than the length of the current bitstring, the bitstring is padded with empty bits. If the new length is less than the length of the current bitstring, the current bitstring is truncated.
* GetInputStream
Returns the bits in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the bitstring contains matching symbols.
* SetInputStream
Sets the value of this BitstringValue by providing a source stream that is used by the BitstringValue object to read the value content. Values set this way don’t contain matching symbols.
* IsMatchingAt
Returns true if the item at position of this TTCN‑3 bitstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* GetMatchingAt
If the position of this TTCN‑3 bitstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* SetMatchingAt
Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.

#### 12.4.4.7 OctetstringValue

**OctetstringValue** is mapped to the following interface:

public interface ITciOctetstringValue : ITciValue {
 string StringValue { get; set; }
 byte this[int position] { get; set; }
 int Length { get; set; }
 System.IO.Stream GetInputStream ();

 void SetInputStream(System.IO.Stream stream);
 bool IsMatchingAt (int position);

 ITciMatchingMechanism GetMatchingAt (int position);

 void SetMatchingAt (int position, ITciMatchingMechanism template);

}

**Members:**

* StringValue
Gets or sets the string value of the TTCN-3 octetstring. The only allowed characters in the string passed to this property are hexadecimal digits. The length of the string passed to this property shall be even. The string returned by this property is a sequence of pairs of hexadecimal digits. In some cases, the message data are available in the form of a byte stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this property returns null and the GetInputStream method shall be used for reading the data.
* Indexing operator
Get or sets the value of the octet at the specified position. IndexOutOfRangeException is thrown if the position is less than zero or greater or equal to the string length.
* Length
Gets or sets the length of this ITciOctetstringValue in octets. The property returns zero if the value of this object is omit. In case the new length is greater than the length of the current octetstring, the octetstring is padded with empty octets. If the new length is less than the length of the current octetstring, the current octetstring is truncated.
* GetInputStream
Returns the octets in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the octetstring contains matching symbols.
* SetInputStream
Sets the value of this OctetstringValue by providing a source stream that is used by the OctetstringValue object to read the value content. Values set this way don’t contain matching symbols.
* IsMatchingAt
Returns true if the item at position of this TTCN‑3 octetstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* GetMatchingAt
If the position of this TTCN‑3 octetstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* SetMatchingAt
Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.

#### 12.4.4.9 HexstringValue

**HexstringValue** is mapped to the following interface:

public interface ITciHexstringValue : ITciValue {
 string StringValue { get; set; }
 byte this[int position] { get; set; }
 int Length { get; set; }
 System.IO.Stream GetInputStream ();

 void SetInputStream(System.IO.Stream stream, int numberOfItems);
 bool IsMatchingAt (int position);

 ITciMatchingMechanism GetMatchingAt (int position);

 void SetMatchingAt (int position, ITciMatchingMechanism template);

}

**Members:**

* StringValue
Gets or sets the string value of the TTCN-3 hexstring. The only allowed characters in the string passed to this property are hexadecimal digits. The string returned by this property is a sequence of hexadecimal digits. In some cases, the message data are available in the form of a stream and cannot be converted into a string by TCI (e.g. because of memory restrictions). In such cases, this property returns null and the GetInputStream method shall be used for reading the data.
* Indexing operator
Get or sets the hex digit at the specified position. Only the lower four bits of the passed value are used in this assignment. The upper four bits as ignored. IndexOutOfRangeException is thrown if the position is less than zero or greater or equal to the string length.
* Length
Gets or sets the length of this ITciHexstringValue in hex digits. The property returns zero if the value of this object is omit. In case the new length is greater than the length of the current hexstring, the hexstring is padded with zeroes. If the new length is less than the length of the current hexstring, the current hexstring is truncated.
* GetInputStream
Returns the content in the form of an input stream. Repeated calls to the same method return different stream instances. The method returns null if the hexstring contains matching symbols.
* SetInputStream
Sets the value of this HexstringValue by providing a source stream that is used by the HexstringValue object to read the value content. Values set this way don’t contain matching symbols.
* IsMatchingAt
Returns true if the item at position of this TTCN‑3 hexstring is a matching mechanism inside a value (AnyElement, AnyElementsOrNone) and false otherwise.
* GetMatchingAt
If the position of this TTCN‑3 hexstring contains a matching mechanism inside a value (AnyElement, AnyElementsOrNone), the method returns it. Otherwise the distinct value null is returned.
* SetMatchingAt
Sets a matching mechanism at position. Only two matching mechanisms are allowed: AnyElement and AnyElementsOrNone.