#### 7.3.2.2 TCI‑CD provided

This clause specifies the operations the TM shall provide to the TE.

##### 7.3.2.2.1 decode

|  |  |
| --- | --- |
| Signature | Value decode(in TriMessageType message, in Type decodingHypothesis) |
| In Parameters | message | The encoded message to be decoded. |
| decodingHypothesis | The hypothesis the decoding can be based on. |
| Return Value | Returns the decoded value, if the value is of a compatible type as the decodingHypothesis, else the distinct value null. |
| Constraint | This operation shall be called whenever the TE has to implicitly decode an encoded value (e.g. when performing a port operation such as receive, trigger, getcall, getreply, catch, check or calling an external function). The TE might decode immediately after reception of an encoded value, or might for performance considerations postpone the decoding until the actual access of the encoded value. |
| Effect | This operation decodes message according to the encoding rules and returns a TTCN‑3 value. The decodingHypothesis shall be used to determine whether the encoded value can be decoded. If an encoding rule is not self‑sufficient, i.e. if the encoded message does not inherently contain its type decodingHypothesis shall be used. If the encoded value can be decoded without the decoding hypothesis, the distinct null value shall be returned if the type determined from the encoded message is not compatible with the decoding hypothesis. |

##### 7.3.2.2.2 encode

|  |  |
| --- | --- |
| Signature | TriMessageType encode(in Value value) |
| In Parameters | value | The value to be encoded. |
| Return Value | Returns an encoded TriMessage for the specified encoding rule. |
| Constraint | This operation shall be called whenever the TE has to implicitly encode a value (e.g. when perfoming a port operation such as send, call, reply and raise or calling an external function) . |
| Effect | Returns an encoded TriMessage according to the encoding rules. |

##### 7.3.2.2.3 decodeValue

|  |  |
| --- | --- |
| Signature | TInteger decodeValue(inout TriMessageType encodedValue, in Type decodingHypothesis, in TString encodingInfo, out Value decodedValue) |
| In Parameters | message | The encoded message to be decoded. |
| decodingHypothesis | The hypothesis the decoding can be based on. |
| decodingInfo | Dynamic decoding parameters |
| Out Parameters | decodedValue | The decoded value, if the value is of a compatible type as the decodingHypothesis, else the distinct value null. |
| Return Value | An integer value indicating success of the operation: 0 in case of success, 1 in case of an uspecified decoding error and 2 if decoding could not be completed because encodedValue didn’t contain enough bits. |
| Constraint | This operation shall be called whenever the TE invokes the predefined functions decvalue or decvalue\_unichar. |
| Effect | This operation decodes message according to the encoding rules and returns the result of the decoding operation. The decodingHypothesis shall be used to determine whether the encoded value can be decoded. If an encoding rule is not self‑sufficient, i.e. if the encoded message does not inherently contain its type decodingHypothesis shall be used. In case of success, the used bits are removed from the encodedValue parameter and the decoded TTCN‑3 value is passed to the caller in the decodedValue parameter. In case of failure, the TE shall ignore the content of the encodedValue and decodedValue parameter and shall act as if the former one were unchanged and the latter one contained the distinct value null. |

##### 7.3.2.2.4 encodeValue

|  |  |
| --- | --- |
| Signature | TriMessageType encode(in Value value, in TString encodingInfo) |
| In Parameters | value | The value to be encoded. |
| encodingInfo | Dynamic encoding parameters |
| Return Value | Returns an encoded TriMessage for the specified encoding rule. |
| Constraint | This operation shall be called whenever the TE invokes the predefined functions encvalue or encvalue\_unichar. |
| Effect | Returns an encoded TriMessage according to the encoding rules. |

#### 8.3.2.15 TciDecodingResult

**TciDecodingResult** is used as a return type of the TciCDProvided.decodeValue operation and it is defined as follows:

package org.etsi.ttcn.tci;

public interface TciDecodingResult {

 public int getResult ();

 public Value getDecodedValue ();

}

**Methods:**

* getResult Returns the numeric result of the TciCDProvided.decodeValue operation
* getDecodedValue Returns the decoded value or the distinct value null.

#### 8.5.2.1 TCI‑CD provided

The TCI‑CD Provided interface is mapped to the following interface:

// TCI‑CD

// TE ‑> CD

package org.etsi.ttcn.tci;

public interface TciCDProvided {

 public Value decode (TriMessage message, Type decodingHypothesis );

 public TriMessage encode (Value value);

 public TciDecodingResult decodeValue (TriMessage message, Type decodingHypothesis,
 String decodingInfo );

 public TriMessage encodeValue (Value value, String encodingInfo);

}

#### 9.4.2.1 TCI‑CD provided

The TCI‑CD Provided interface is mapped to the following interface:

Value tciDecode (BinaryString message, Type decHypothesis)

BinaryString tciEncode(Value value)

int tciDecodeValue (BinaryString \* message, Type decHypothesis, String decodingInfo, Value \* decodedValue)

BinaryString tciEncodeValue(Value value, string encodingInfo)

NOTE: BinaryString type reused from TRI.

#### 10.6.2.2 TciCdProvided

This class defines the TCI\_CD provided interface. It is mapped to the following interface:

//Destructor

virtual ~TciCdProvided ();

//This operation is called whenever the TE has to decode a value in a port operation

virtual TciValue \* decode (const TriMessage \*p\_message, const TciType \*p\_decodingHypothesis)=0;

//This operation is called whenever the TE has to encode a value in a port operation

virtual TriMessage \* encode (const TciValue \*p\_value)=0;

//This operation is called whenever the TE invokes decvalue

virtual Tinteger decodeValue (TriMessage \*p\_message, const TciType \*p\_decodingHypothesis,

 const TuniversalString & decodingInfo, TciValue \*\* decodedValue )=0;

//This operation is called whenever the TE invokes encvalue

virtual TriMessage \* encodeValue (const TciValue \*p\_value, const TuniversalString & encodingInfo)=0;

#### 12.5.2.1 TCI-CD provided

The **TCI-CD provided** interface is mapped to the following interface:

public interface ITciCDProvided {
 ITciValue Decode(ITriMessage message,
 ITciType decodingHypothesis);
 ITriMessage Encode(ITciValue value);
 int DecodeValue(ITriMessage message,
 ITciType decodingHypothesis,

 string decodingInfo,

 out ITciValue decodedValue);
 ITriMessage EncodeValue(ITciValue value,

 string encodingInfo);
}