

Implementation of
ISO/IEC 8882-1:1993

Information technology — Telecommunications and information exchange between systems — X.25 DTE conformance testing —

Part 1: General principles

UDC 681.327:621.39

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW



Document Sharing Hub

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/6, upon which the following bodies were represented:

- British Computer Society
- British Telecommunications plc
- CCTA (Government Centre for Information Systems)
- Digital Equipment Co. Ltd.
- IBM United Kingdom Ltd.
- Information Systems Committee of the Universities Funding Council
- Institution of Electrical Engineers
- International Computers Limited
- Logica UK Ltd.
- Ministry of Defence
- Rank Xerox Ltd.
- Swift Project Consortium

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Standards Board and comes into effect on 15 March 1994

© BSI 05-2000

The following BSI references relate to the work on this standard:
Committee reference IST/6
Draft for comment 91/62756 DC

ISBN 0 580 23010 4

Amendments issued since publication

Amd. No.	Date	Comments

Contents

	Page
Committees responsible	Inside front cover
National foreword	ii
Foreword	iii
Text of ISO/IEC 8882-1	1

National foreword

This British Standard reproduces verbatim ISO/IEC 8882-1:1993 and implements it as the UK national standard.

This British Standard is published under the direction of the Information Systems Technology Standards Policy Committee whose Technical Committee IST/6 has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international committee any enquiries on interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

NOTE International and European Standards, as well as overseas standards, are available from Customer Services, Publications, BSI, Linford Wood, Milton Keynes, MK14 6LE.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the ISO/IEC title page, pages ii to iv, pages 1 to 91 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

**Information technology —
Telecommunications and information
exchange between systems — X.25 DTE
conformance testing —**

Part 1:
General principles

*Technologies de l'information — Télécommunications et échange
d'information entre systèmes — Test de conformité X.25 DTE —
Partie 1: Principes généraux*



Reference number
ISO/IEC 8882-1:1993(E)

Contents

	Page
Foreword	iii
Introduction	1
1 Scope	1
2 Normative references	1
3 Definitions	2
4 Abbreviations	2
5 Test notation	3
6 Test suite structure	3
7 Testing methodology	3
7.1 Test principles	3
7.2 Data transfer	3
7.3 Other user data fields	4
7.4 Testing configuration	4
7.5 Operational consideration	4
7.6 DTE initiated actions	4
7.7 Timing considerations	4
7.8 Optional facility testing	5
7.9 Transient states	6
8 Structure of other parts of ISO/IEC 8882	6
Annex A (informative) DIS level text ISO/IEC 9646-3	7
Figure 1 — Data Link Layer Data Transfer Configuration	4
Figure 2 — Packet Layer Data Transfer Configuration	5
Figure 3 — Example of the use of a Packet Layer Sub-function	5
Figure 4 — Example of the use of a Transport Layer Sub-function directly over ISO/IEC 8208 (i.e. OSI Network Layer)	5
Figure A.1 — T_CONNECTrequest Abstract Service Primitive	35
Figure A.2 — A PDU Constraints Table that Uses a Field Group	48
Figure A.3 — Field Group A's Constraint A1 in the Single Form	49
Figure A.4 — Field Group A's Constraint A2 in the Single Form	49
Figure A.5 — Field Group A's Constraints A1 and A2 in the Multiple Form	49
Figure A.6 — Test Case Behaviour Structure	60
Figure A.7 — Structured Test Case Behaviour	61
Figure A.8 — Principle of a TTCN Behaviour Tree	66
Figure A.9 — A TTCN Behaviour Tree	67
Table A.1 — The TTCN.MP Syntactic Metanotation	13
Table A.2 — Predefined Character String Types	20
Table A.3 — Precedence of Operators	24
Table A.4 — Units of Length Used in Field Length Specifications	52

Descriptors: Data processing, information interchange, data transmission, data processing equipment, data terminal equipment, tests, comparative tests, generalities.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8882-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-Committee SC 6, *Telecommunications and information exchange between systems*.

ISO/IEC 8882 consists of the following parts, under the general title *Information technology — Telecommunications and information exchange between systems — X.25 DTE conformance testing*:

- *Part 1: General principles*;
- *Part 2: Data link layer conformance test suite*;
- *Part 3: Packet level conformance test suite*.

Annex A forms an integral part of ISO/IEC 8882.

Introduction

ISO/IEC 8882 specifies a set of tests to evaluate Data Terminal Equipment (DTE) conformance to International Standards ISO 7776 or ISO/IEC 8208, or both. ISO 7776 and ISO/IEC 8208 allow for a DTE to interface with a Data Circuit-Terminating Equipment (DCE) conforming to CCITT Recommendation X.25 (1980, 1984) or to another DTE conforming to ISO 7776 or ISO/IEC 8208 or both. The implementations of ISO 7776 and ISO/IEC 8208 are tested independently.

CCITT Recommendations X.25(1980) and X.25(1984) are written from the perspective of a DCE and therefore do not explicitly specify the DTE operation. However, recommended operation of DTEs is included by implication because of the need to communicate with X.25 DCEs. Tests within ISO/IEC 8882-2 and ISO/IEC 8882-3 pertaining to X.25 (1980, 1984) are based on the DTE operational characteristics implied by CCITT X.25.

This part of ISO/IEC 8882 specifies the framework in which the other parts of ISO/IEC 8882 may be understood and the principles to be applied. The notation used in ISO/IEC 8882-2 and ISO/IEC 8882-3 is TTCN as defined in ISO/IEC 9646-3.

ISO/IEC 8882-2 presents the Data Link Layer aspects for evaluating conformance to ISO 7776 while ISO/IEC 8882-3 presents the Packet Layer aspects for evaluating conformance to ISO/IEC 8208.

The conformance tests are designed for use by:

- test evaluators (responsible for analysing results and determining whether conformance has been achieved);
- test suite designers or implementors (for determining what tests are required and what results can and should be anticipated by the test device); and
- users implementing ISO 7776 or ISO/IEC 8208 or DTEs interfacing to DCEs that implement CCITT X.25 (1980 or 1984) (for determining the functionality required of their implementations to be considered in conformance).

1 Scope

ISO/IEC 8882 defines the testing of a DTE operating at the Data Link Layer and at the Packet Layer when accessing, by means of a dedicated path connection, switched or permanent, a public or private packet-switched network conforming to CCITT Recommendation X.25 or another DTE conforming to ISO 7776 and ISO/IEC 8208.

The tests will test the conformance of an implementation by observing its external behaviour. The conformance tests will not test the DTE performance characteristics, the diagnostic and maintenance functions, the correctness of the protocol itself, or DTE internal implementation, or the full capabilities as stated in the PICS.

This part of ISO/IEC 8882

- provides a general introduction;
- refers to those applicable International Standards;
- defines terms applicable to X.25-DTE conformance testing;
- states the test case derivation and description; and
- states the test methodology.

ISO/IEC 8882-1 contains no statement of conformance. Specific statements of conformance are given in ISO/IEC 8882-2 and ISO/IEC 8882-3.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 8882. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 8882 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498:1984, *Information processing systems — Open Systems Interconnection — Basic Reference Model*.

ISO 7776:1986, *Information processing systems — Data communications — High-level data link control procedures — Description of the X.25 LAPB-compatible DTE data link procedures*.

ISO/IEC 8208:1990, *Information technology — Data communications — X.25 Packet Layer Protocol for Data Terminal Equipment*.

ISO/IEC 8882-2:1992, *Information technology — Telecommunications and information exchange between systems — X.25 DTE conformance testing — Part 2: Data link layer conformance test suite*.

ISO/IEC 8882-3:1991, *Information technology — Telecommunications and information exchange between systems — X.25 DTE conformance testing — Part 3: Packet layer conformance test suite*.

ISO/IEC 9646-1:1991, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts*. [See also CCITT Recommendation X.291 (1991)].

ISO/IEC 9646-2:1991, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 2: Abstract test suite specification*. [See also CCITT Recommendation X.291 (1991)].

ISO/IEC 9646-3:1992, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 3: The Tree and Tabular Combined Notation (TTCN)*.

CCITT Recommendation X.25 (1980), *Interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks*.

CCITT Recommendation X.25 (1984), *Interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit*.

3 Definitions

3.1 Reference model definitions

This part of ISO/IEC 8882 makes use of the following term defined in ISO 7498:

(N)-protocol-data-unit (N)-PDU

3.2 Conformance testing definitions

This part of ISO/IEC 8882 makes use of the following terms defined in ISO/IEC 9646-1:

- a) Abstract Test Case
- b) Conformance Test Suite
- c) Conformance Testing
- d) Implementation Under Test
- e) Inopportune PDU
- f) Lower Tester
- g) Protocol Implementation Conformance Statement
- h) Protocol Implementation eXtra Information for Testing
- i) Remote Single Layer Test Method
- j) System Under Test
- k) Test Group
- l) Test Step
- m) Test Suite

3.3 X.25 DTE conformance testing definitions

For the purpose of this part of ISO/IEC 8882 the following definitions apply:

3.3.1 improper PDU

the (N)-PDU whose syntax does not conform to the format specifications of ISO 7776 or ISO/IEC 8208 or CCITT X.25

3.3.2 proper PDU

the (N)-PDU whose syntax conforms to the format specification of CCITT X.25, ISO 7776 or ISO/IEC 8208 and is acceptable to the state or phase of the interface

3.3.3 tester

refer to Lower Tester

3.3.4 test case

refer to Abstract Test Case

3.3.5 test selection

test selection is the process of choosing test cases according to the specific criteria based on the IUT's PICS and PIXIT in order to constitute a conformance test suite for the IUT

3.3.6 test subgroup

a set of test cases that share a common characteristic, such as testing for proper, improper, or inopportune PDUs. A test subgroup is the smallest testable set of test cases that can be selected

3.3.7 sub-function

a subset of the PDUs and functional capabilities of the protocol level above the IUT that are needed to allow data transfer testing to be accomplished

4 Abbreviations

The following abbreviations are used in this part of ISO/IEC 8882:

DCE Data Circuit-Terminating Equipment
DTE Data Terminal Equipment
DXE DTE or DCE
IUT Implementation Under Test
PDU Protocol Data Unit
PICS Protocol Implementation Conformance Statement

PIXIT Protocol Implementation eXtra
 Information for Testing
 RS Remote Single Layer
 SUT System Under Test
 TPDU Transport Protocol Data Unit
 TTCN Tree and Tabular Combined Notation

5 Test notation

The test notation used in ISO/IEC 8882-2 and ISO/IEC 8882-3 is TTCN as defined in the DIS version of ISO/IEC 9646-3. This version of ISO/IEC 9646-3 is contained in Annex A. ISO/IEC 8882-2 and ISO/IEC 8882-3 contain an annex describing the differences between the DIS version of TTCN used and the version of TTCN defined in ISO/IEC 9646-3.

6 Test suite structure

The test suite structure used in ISO/IEC 8882-2 and ISO/IEC 8882-3 is defined in ISO/IEC 9646-2 and is illustrated below.

Test Suite Structure

```

Test Group
  Test Subgroup 1 (Proper PDUs)
    Test Case No.101
    Test Case No.102
    .
    .
    .
    Test Case No.1nn
  Test Subgroup 2 (Improper PDUs)
    Test Case No.201
    Test Case No.202
    .
    .
    .
    Test Case No.2nn
  Test Subgroup 3 (Inopportune PDUs)
    Test Case No.301
    Test Case No.302
    .
    .
    .
    Test Case No.3nn
  
```

7 Testing methodology

The testing methodology is based on the OSI Conformance Testing Methodology and Framework. The test method used is the Remote Single layer (RS) method. To employ the RS method effectively, the concept of using sub-functions of higher layer protocols is introduced. Sub-functions are a subset of the PDUs and functional capabilities of the protocol layer above the IUT that are needed to allow data transfer testing to be accomplished. The required properties of the sub-functions used are:

- That the number and sequence of data-PDUs received from the IUT after receiving a data-PDU from the tester is predictable, and that the number received from the IUT is greater than zero.
- That the reactions of the IUT upon receipt of these data-PDUs are known.
- That the sub-function allows either the tester or IUT to initiate transmission of the data-PDUs.
- That the sub-function allows for the exchange of data-PDUs by the layer under test with minimal interference from other functions of the protocol layer(s) above the IUT (e.g., PDU retransmission, error recovery, etc.).

Examples of data transfer configurations are shown for the Data Link Layer and the Packet Layer in Figure 1 and Figure 2 respectively.

NOTE The requirements on underlying protocols are specified in ISO/IEC 8208, clause 3.

7.1 Test principles

The testing of the Data Link and the Packet Layer protocols is done separately. The data link layer is normally tested first since the packet layer requires the correct operation of the data link layer. The RS method is the selected test method since it cannot be assumed that a tester will be able to test completely each level as a separate entity. The RS method requires that the tester shall recognize and respond to a PDU received from the higher level protocols. The specific PDUs which shall be accepted are defined in ISO/IEC 8882-2 and ISO/IEC 8882-3.

7.2 Data transfer

The sub-function chosen by the IUT provider should create an alternating exchange of data-PDUs between the IUT and the tester. This exchange will be repeated until the sequence numbers of the layer under test have been rotated. The sub-function chosen shall be defined in the PIXIT of the IUT, and shall include the sequence and contents of the user data fields required for the test. Two examples of the use of a sub-function to accomplish data transfer testing are shown in Figure 3 and Figure 4.

A more detailed explanation of data transfer testing is provided in ISO/IEC 8882-2 and ISO/IEC 8882-3. These explanations also address the data transfer testing of send-only and receive-only IUTs.

It is recognized that an IUT provider may not be able to accomplish data transfer testing by this means. In such instances the data transfer tests are deselected.

7.3 Other user data fields

When necessary, the content of user data fields in other than data-PDUs shall be provided to the tester by the owner of the IUT in order to execute successfully the conformance test suite. In this case, the IUT requires the tester to transmit user data fields in accordance with higher level protocols which are operating above the IUT. For example, user data fields of call set-up, clearing, and interrupt packets of the Packet Layer may be affected.

The content of such user data fields shall be provided by the IUT owner in the PIXIT.

Sub-function of Packet Layer	Sub-function of Other Protocol(s)
Data Link Layer (layer under test)	

Figure 1 — Data Link Layer Data Transfer Configuration

7.4 Testing configuration

The SUT is connected to the tester, point-to-point, when participating in active testing. The points of observation and control for each test sequence are within the tester.

ISO/IEC 8882-2 and ISO/IEC 8882-3 include PIXIT proformas which, when completed, describe the dynamic conformance test environment.

7.5 Operational consideration

Testing is done in a controlled environment. It is not the intent of this document to define the operational characteristics of test devices used to achieve DTE Conformance Testing. However, it is highly desirable that the device be capable of segregating IUT test activity from normal operation of underlying layers. At a minimum, the tester should be capable of distinguishing between I-frame retransmission at the data link layer (due to T1 expiration) and a packet layer retransmission. Some recommended functions of the tester include:

- a) Detection of failures of the physical layer.
- b) The ability to respond transparently to timeout conditions at the data link layer.

c) Timely data link layer acknowledgement (to avoid retransmissions) when performing packet layer testing.

d) In the instance where an I-frame is retransmitted, the tester should properly acknowledge the frame and not pass it on to the packet layer. The tester shall be sensitive to failures that interfere with the tests, and when such a condition is detected, the tester should abort the test.

e) The tester shall recognise the possibility of receiving unexpected PDUs which do not affect the results of the test case. These specific PDUs for each layer are defined in ISO/IEC 8882-2 and ISO/IEC 8882-3. In addition, other unexpected PDUs may be received which do affect the results of a test case. These PDUs will require further analysis and potentially, re-execution of the test case. Receipt of such PDUs may be due to interference from sources outside the realm of the X.25 environment (e.g. the IUT operating system, IUT operator).

7.6 DTE initiated actions

Generally the tester forces the IUT to transmit a particular PDU. However, in order to execute some test groups, it is required that the IUT initiate the transmission of particular PDUs. When a DTE-initiated action is required, it is specified in the appropriate test group. Direct control of such actions may not be feasible for the IUT owner. In such instances these tests are deselected.

7.7 Timing considerations

There are two types of timing considerations which should be taken into account — timing considerations for the tester and timing considerations for the SUT.

a) Tester Considerations: The tester shall allow for the time required by the IUT to progress from one test case to the next. This timing consideration should be accomodated for in the test preamble.

For example, the time required by the IUT to initiate a CALL REQUEST after completing a CALL CLEARING operation, and the time required by the IUT to re-establish the data link after completing a disconnect operation. The precise timing requirements of the IUT shall be specified in the PIXIT, as defined in ISO/IEC 8882-2 and ISO/IEC 8882-3.

b) SUT Considerations: Where the protocol standard identifies a need for timers, values for those timers shall be those stated in the PIXIT.

Sub-function of OSI Protocol(s)		Sub-function of non-OSI Protocol(s)
Packet Layer (layer under test)		
Data Link Layer	LAN Protocol(s)	Other Protocol(s)
Note	Note	Note

Figure 2 — Packet Layer Data Transfer Configuration

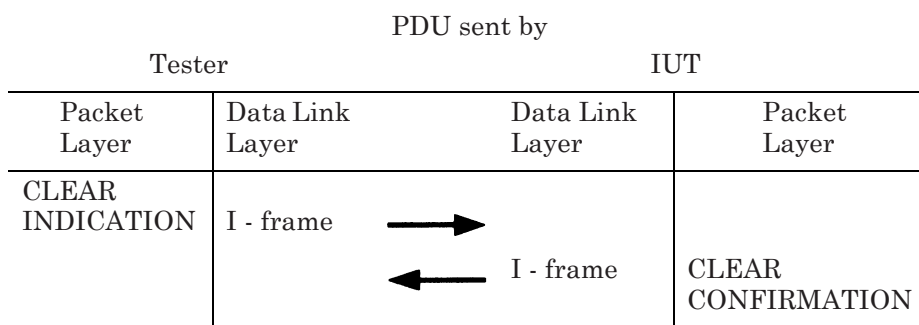


Figure 3 — Example of the use of a Packet Layer Sub-function

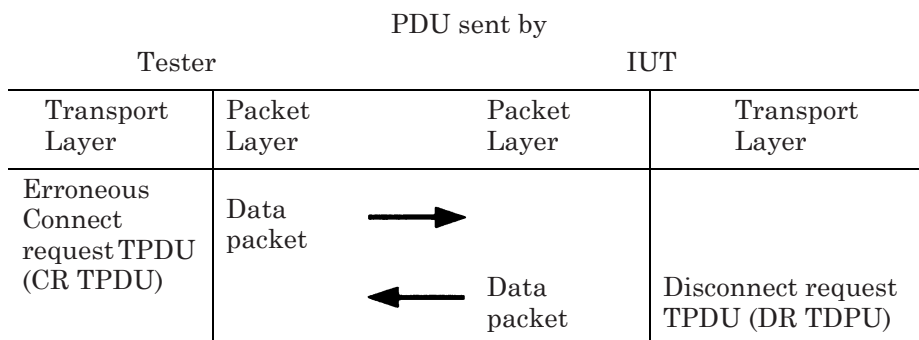


Figure 4 — Example of the use of a Transport Layer Sub-function directly over ISO/IEC 8208 (i.e. OSI Network Layer)

7.8 Optional facility testing

Full testing of optional facilities is not possible because

- a) optional facilities may be managed by levels above X.25; and
- b) multiple combinations of optional facilities may be required depending on the applications running above X.25.

Optional facilities are tested individually. Where the IUT cannot support this method of testing these tests are deselected.

7.9 Transient states

It is recognized that for those IUTs that process PDUs sequentially, certain states are not realizable. Specifically, the testing of the IUT during the DXE defined states (for example, for the packet layer, r3 — Restart Indication, p3 — Incoming Call, p7 — Clear Indication, and d3 — Reset indication) may result in the testing of some other states (p1 — Ready, p4 — Data Transfer, d1 — Flow Control Ready). For example, to test the response to an error packet (inopportune or improper packet) in the DXE Restart Indication (r3) state, the tester will send a Restart Indication, immediately followed by the error packet. The tester expects the IUT to discard the error packet and then send a Restart Request in response to the error packet. However, the IUT generally responds immediately to the Restart Indication with a Restart Confirmation and processes the next packet from the packet level state r1. When these states are not observable in the IUT, transient test cases are deselected. The specific handling of transient state testing is described in ISO/IEC 8882-2 and ISO/IEC 8882-3.

8 Structure of other parts of ISO/IEC 8882

In order to ensure consistency between ISO/IEC 8882-2 and ISO/IEC 8882-3 the following items shall be included in those standards.

- a) A PIXIT pro forma
- b) A statement of Acceptable Unexpected responses.
- c) A statement of Tester Tuning Considerations.
- d) PICS and PIXIT based abstract test selection rules.
- e) A definition of the test cases.
- f) A statement of conformance.
- g) An annex describing the differences between the version of TTCN used and the version of TTCN defined in ISO/IEC 9646-3.

Annex A (informative)

DIS level text for ISO/IEC 9646-3

The Tree and Tabular Combined Notation (TTCN)

Important — This Annex contains an extract from the DIS text of ISO/IEC 9646-3. The clause, figure and table numbering has been changed to align with the numbering in this standard. Example and proforma numbering is unchanged from the original text. References to the annexes in the original text have been placed in braces; for example “{Annex A}”. All errors contained in the original text which were subsequently corrected in the published International Standard are also present here.

Introduction

This Part of the multi-part “standard/recommendation” (hereafter abbreviated to “standard*”) defines a test notation, called the Tree and Tabular Combined Notation (TTCN), for use in the specification of “OSI or related CCITT X.series or T.series” (hereafter abbreviated to “OSI*”) generic and abstract conformance test suites.

In constructing a generic or abstract test suite, a test notation is used to describe abstract test cases. The test notation can be an informal notation (without formally defined semantics) or a formal description technique (FDT). TTCN is an informal notation with clearly defined, but not formally defined, semantics.

TTCN is designed to meet the following objectives:

- a) to provide a notation in which generic and abstract test cases can be expressed in test suite standards*;
- b) to provide a notation which is independent of test methods, layers and protocols;
- c) to provide a notation which reflects the abstract testing methodology defined in this multi-part standard*.

In the abstract testing methodology a test suite is looked upon as a hierarchy ranging from the complete test suite, through test groups, test cases and test steps, down to test events. TTCN provides a naming structure to reflect the position of test cases in this hierarchy. It also provides the means of structuring test cases as a hierarchy of test steps culminating in test events. In TTCN the basic test events are sending and receiving Abstract Service Primitives (ASPs), Protocol Data Units (PDUs) and timer events.

Two forms of the notation are provided: a human-readable tabular form, called TTCN.GR, for use in OSI* conformance test suite standards; and a machine-processable form, called TTCN.MP, for use in representing TTCN in a canonical form within computer systems and as the syntax to be used when transferring TTCN test cases between different computer systems. The two forms are semantically equivalent.

A.1 Scope

This Part of the multi-part standard* defines an informal test notation, called TTCN, for OSI* conformance test suites, which is independent of test methods, layers and protocols, and which reflects the abstract testing methodology defined in Parts 1 and 2 of this multi-part standard*.

It also specifies requirements and provides guidance for using TTCN in the specification of system-independent conformance test suites for one or more OSI* standards*. It specifies two forms of the notation: one, a human-readable form, applicable to the production of conformance test suite standards* for OSI* protocols; and the other, a machine-processable form, applicable to processing within and between computer systems.

This Part of this multi-part standard* applies to the specification of conformance test cases which can be expressed abstractly in terms of control and observation of protocol data units and abstract service primitives. Nevertheless, for some protocols, test cases may be needed which cannot be expressed in these terms. The specification of such test cases is outside the scope of this standard*, although those test cases may need to be included in a conformance test suite standard*.

NOTE 1 For example, some static conformance requirements related to an application service may require testing techniques which are specific to that particular application.

This Part of this multi-part standard* applies to the specification of conformance test suites for OSI* protocols in OSI layers 2 to 7, specifically including ASN.1 based protocols. The specification of conformance test suites for multi-peer or Physical layer protocols is outside the scope of this standard*.

The relation between TTCN and formal description techniques is outside the scope of this standard*.

The specification of test cases in which more than one behaviour description is to be run concurrently, is outside the scope of this standard*.

NOTE 2 Use of parallel trees and synchronization between them is expected to be covered by an Addendum to this standard*.

Although this Part of this multi-part standard* specifies requirements on abstract test suites written in TTCN, including their operational semantics, the means of realization of executable test suites from abstract test suites is outside the scope of this Part. Nevertheless, this Part specifies requirements on what a test suite standard* may specify about a conforming realization of the test suite.

NOTE 3 ISO 9646-4 specifies requirements concerning test realization including ETS derivation.

A.2 Normative References

ISO 9646-1, *Information Processing Systems — Open Systems Interconnection — OSI Conformance Testing Methodology and Framework — Part 1: General Concepts*. (See also CCITT Recommendation X.290)

ISO 9646-2, *Information Processing Systems — Open Systems Interconnection — OSI Conformance Testing Methodology and Framework — Part 2: Abstract Test Suite Specification*. (See also CCITT Recommendation X.290)

ISO 646, *Information Processing Systems — Open Systems Interconnection — ISO 7-bit Coded Character Set for Information Exchange*.

ISO 8824:1989, *Information Processing Systems — Open Systems Interconnection — Abstract Syntax Notation One (ASN.1)*. (See also CCITT Recommendation X.208)

ISO 8825:1989, *Information Processing Systems — Open Systems Interconnection — Basic Encoding Rules for ASN.1*. (See also CCITT Recommendation X.209)

NOTE These versions of ASN.1 include ASN.1 Extensions Addenda.

ISO 7498-1, *Information Processing Systems — Open Systems Interconnection — Basic Reference Model*. (See also CCITT Recommendation X.200)

ISO TR 8509, *Information Processing Systems — Open Systems Interconnection — Service Conventions*. (See also CCITT Recommendation X.210)

A.3 Definitions

A.3.1 Basic Terms from ISO 9646-1

The following terms defined in ISO 9646-1 apply:

- a) abstract service primitive
- b) abstract testing methodology
- c) abstract test case
- d) abstract test method
- e) abstract test suite
- f) conformance log
- g) conformance statement
- h) conformance testing
- i) conformance test suite
- j) coordinated test method
- k) distributed test method
- l) embedded testing
- m) executable test case
- n) executable test suite
- o) external test methods
- p) fail verdict
- q) foreseen outcome
- r) generic test case
- s) generic test suite
- t) implementation under test
- u) inconclusive verdict

- v) inopportune test event
- w) local test methods
- x) lower tester
- y) multi-layer testing
- z) pass verdict
- aa) PICS proforma
- ab) PIXIT proforma
- ac) point of control and observation
- ad) protocol data unit
- ae) protocol implementation
- af) real tester
- ag) remote test method
- ah) syntactically invalid test event
- ai) system under test
- aj) test case
- ak) test coordination procedures
- al) test event
- am) test group
- an) test management protocol
- ao) test outcome
- ap) test purpose
- aq) test realizer
- ar) test step
- as) test suite
- at) unforeseen outcome
- au) upper tester
- av) valid test event
- aw) verdict

A.3.2 Terms from ISO 7498-1

The following terms defined in ISO 7498-1 apply:

- a) (N)-layer
- b) (N)-protocol
- c) (N)-protocol control information
- d) (N)-protocol data unit
- e) (N)-service
- f) (N)-service access point
- g) (N)-user data transfer syntax

A.3.3 Terms from ISO TR 8509

The following terms defined in ISO TR 8509 apply:

- a) service primitive
- b) service provider
- c) service user

A.3.4 Terms from ISO 8825

The following term defined in ISO 8825 applies:

- encoding

A.3.5 Terms from ISO 8824

The following terms defined in ISO 8824 apply:

- a) NumericString
- b) PrintableString
- c) TeletexString
- d) VideotexString
- e) VisibleString
- f) IA5String
- g) GraphicString
- h) GeneralString

A.3.6 TTCN Specific Terms

For the purposes of this standard* the following definitions apply:

A.3.6.1 Abbreviation identifier: A name for an abbreviation, which identifies its definition.

A.3.6.2 Attach statement: A TTCN statement which attaches a sub-tree to a calling tree.

A.3.6.3 Base constraint: Specifies a set of default values for each and every field in an ASP or PDU type declaration.

A.3.6.4 Behaviour line: An entry in a dynamic behaviour table representing a test event or other TTCN statement together with associated label, verdict, constraints reference and comment information as applicable.

A.3.6.5 Behaviour tree: A specification of a set of sequences of test events, and other TTCN statements.

A.3.6.6 Blank entry: In a modified multiple constraint a blank entry in a constraint parameter or field denotes that a constraint value is to be inherited.

A.3.6.7 Calling tree: The behaviour tree to which a sub-tree is attached.

A.3.6.8 Constraints part: That component of a TTCN test suite concerned with the specification of the values of ASP parameters end parameter groups and PDU fields and field groups.

A.3.6.9 Constraints reference: A reference to a constraint, given in a behaviour line.

A.3.6.10 Decode expression: A specification of the decoding of PDUs embedded in ASPs or other PDUs.

A.3.6.11 Default behaviour: The events, and other TTCN statements, which may occur at any level of the associated tree, and which are indicated in the default behaviour proforma.

A.3.6.12 Defaults library: The set of the default behaviours in a test suite.

A.3.6.13 Defaults reference: A structured name which specifies the location of the default in the defaults library.

A.3.6.14 Dotted identifier: An identifier, consisting of a base constraint identifier concatenated with one or more modified constraint identifiers, separated by dots.

A.3.6.15 Encode expression: A specification of the encoding of PDUs embedded in ASPs or other PDUs.

A.3.6.16 Field groups: A collection of one or more PDU fields which may occur in more than one PDU type declaration and which is defined in a separate declaration.

A.3.6.17 Implicit send event: A mechanism used in Remote methods for specifying that the IUT should be made to initiate a particular PDU or ASP.

A.3.6.18 Inheritance: The means by which constraint values specified for a base constraint are passed to a modified constraint.

A.3.6.19 Local tree: A behaviour tree defined in the same proforma as its calling tree.

A.3.6.20 Modified constraint: A subsequent constraint defined for an ASP or a PDU that already has a Base constraint, and which makes modifications on that Base constraint.

A.3.6.21 Multiple constraint: Declaration of a set of constraints for an ASP or PDU of a given type arranged in a single table.

A.3.6.22 Operational semantics: Semantics explaining the execution of a TTCN behaviour tree.

A.3.6.23 Otherwise event: The TTCN mechanism for dealing with unforeseen events in a controlled way.

- A.3.6.24 Parameter groups:** A collection of one or more ASP parameters which may occur in more than one ASP type declaration and which is defined in a separate declaration.
- A.3.6.25 Pseudo-event:** A pseudo-event is a TTCN expression or Timer operation appearing in the behaviour description.
- A.3.6.26 Receive event:** The receipt of an ASP or PDU at a named or implied PCO.
- A.3.6.27 Root tree:** The main behaviour tree of a test case, occurring at the level of entry into the test case.
- A.3.6.28 Send event:** The sending of an ASP or PDU to a named or implied PCO.
- A.3.6.29 Set of alternatives:** TTCN statements coded at the same level of indentation and belonging to the same predecessor node. They represent the possible events, pseudo-events and constructs which are to be considered at the relevant point in the execution of the test case.
- A.3.6.30 Single constraint:** Declaration of a constraint for a single ASP or PDU of a given type arranged in a single table.
- A.3.6.31 Snapshot semantics:** A semantic model to minimize the effect of timing on the execution of a test case, defined in terms of “snap-shots” of the test environment, during which the environment is effectively frozen for a prescribed period.
- A.3.6.32 Static chaining:** The linking from the declaration of an ASP parameter or PDU field to the declaration of another ASP or PDU.
- A.3.6.33 Static semantics:** Semantic rules that restrict the usage of the TTCN syntax.
- A.3.6.34 Sub-tree:** An identifiable part of a behaviour tree which can be separated, then attached and executed at various places in that (or some other) behaviour tree.
- A.3.6.35 Test case identifier:** A short name for the test case.
- A.3.6.36 Test case reference:** A full name for the test case behaviour description, which defines its conceptual location in the test suite structure.
- A.3.6.37 Test case variable:** One of a set of variables declared globally to the test suite, but whose value is retained only for the execution of a single test case.
- A.3.6.38 Test step library:** The set of the test step dynamic behaviour descriptions in the test suite.
- A.3.6.39 Test step objective:** An informal statement of what the test step is meant to accomplish.
- A.3.6.40 Test Suite constant:** One of a set of constants, *not* derived from the PICS or PIXIT, which will remain constant throughout the test suite.
- A.3.6.41 Test suite parameter:** One of a set of constants derived from the PICS or PIXIT which globally parameterize a test suite.
- A.3.6.42 Test suite variable:** One of a set of variables declared globally to the test suite, and which retain their values between test cases.
- A.3.6.43 Timeout event:** An event which is used within a behaviour tree to check for expiration of a specified timer.
- A.3.6.44 Tree attachment:** The method of indicating that a behaviour tree specified elsewhere (either at a different point in the current proforma, or as a test step in the test step library) is to be included in the current behaviour tree.
- A.3.6.45 Tree header:** That which prefixes a local behaviour tree. The header contains a tree identifier, and a specification of any parameters and their types used in the tree.
- A.3.6.46 Tree identifier:** A name identifying a local behaviour tree.
- A.3.6.47 Tree indentation:** A method of indicating the tree structure of a behaviour description. It is reflected in the behaviour description by indentation of text.
- A.3.6.48 Tree leaf:** The TTCN statement in a behaviour tree or sub-tree which has no specified subsequent behaviour.
- A.3.6.49 Tree node:** A single TTCN statement.
- A.3.6.50 Tree notation:** The notation used in TTCN to represent test cases as trees.
- A.3.6.51 TTCN abbreviation:** A method of indicating a textual substitution to be performed in a dynamic behaviour table.

A.3.6.52 TTCN statement: A TTCN statement is an event, a pseudo-event or construct which is specified in a behaviour description.

A.3.6.53 Unforeseen test event: A test event which has not been identified as a possible outcome in the test suite. It is normally handled using the OTHERWISE event.

A.3.6.54 Unqualified send event: A send event that does not have a Boolean expression or EncodedAs expression on the same statement line.

A.4 Abbreviations

A.4.1 Abbreviations Defined in ISO 9646-1

For the purposes of this Part of ISO 9646, the following abbreviations defined in clause 4 of ISO 9646-1 apply:

ASP: abstract service primitive

OSI: open systems interconnection

OSI*: OSI related CCITT X.series or T.series

PCO: point of control and observation

PDU: protocol data unit

PICS: protocol implementation conformance statement

PIXIT: protocol implementation extra information for testing

SAP: service access point

standard*: standard or recommendation

SUT: system under test

A.4.2 Abbreviations Defined in ISO 9646-2

For the purposes of this Part of ISO 9646, the following abbreviations defined in clause 4 of ISO 9646-2 apply:

DS: distributed single-layer (test method)

FDT: formal description technique

TTCN: tree and tabular combined notation

A.4.3 Other Abbreviations

For the purposes of this Part of ISO 9646, the following abbreviations also apply:

ASN.1: abstract syntax notation one

BNF: The extended Backus-Naur form used in TTCN

CEId: connection endpoint identifier

FIFO: first in first out

TTCN.GR: tree and tabular combined notation, graphical form

TTCN.MP: tree and tabular combined notation, machine processable form

A.5 The Syntax Forms of TTCN

TTCN is provided in two forms:

- a) a graphical form (TTCN.GR) suitable for human readability;
- b) a machine processable form (TTCN.MP) suitable for transmission of TTCN descriptions between machines and possibly suitable for other automated processing.

TTCN.GR is defined using tabular proformas. TTCN.MP differs from TTCN.GR only in syntax; keywords, instead of boxes, are used as information delimiters. The syntax of TTCN.MP is defined in Annex A of this standard* by means of a BNF grammar.

As an aid to clarifying the TTCN.GR many TTCN.MP productions are embedded in the text of this standard*, and are marked: SYNTAX DEFINITION. To improve the readability of this document some productions will appear in the text in several places.

The two forms of TTCN are equivalent. If there is a conflict between the two forms, this is an error, and should be reported back to the standards* organization via a defect report. In such cases, however, the TTCN.MP shall take precedence over the TTCN.GR form pending correction by the standards* organization.

A.6 Compliance

6.1 Test suites that claim to comply with this Part of this multi-part standard* shall state that they comply with the requirements for either TTCN.GR or TTCN.MP.

6.2 Test suites that claim to comply with the requirements of TTCN.GR shall comply with the TTCN.GR syntax requirements stated in clauses A.8 through A.16 and clause {A.3}. Generic test suites may also use the options specified in clause A.18.

6.3 Test suites that claim to comply with the requirements of TTCN.MP shall comply with the TTCN.MP syntax requirements stated in clause {A.3}.

6.4 Test suites that claim to comply with this Part of this multi-part standard* shall comply with the static semantic requirements specified in clauses A.8 through A.15 and have operational semantics in accordance with the definition of the operational semantics in {Annex B}, such that they are semantically valid.

6.5 A test suite standard* that claims to comply with this Part of this multi-part standard* shall require that any realization of that test suite that claims to conform to that test suite standard* shall:

- a) have operational semantics equivalent to the operational semantics of the test suite as defined by {Annex B};
- b) be able to produce a conformance log that as a minimum meets all the requirements in clause 17;
- c) comply with ISO 9646-4.

A.7 Conventions

A.7.1 Introduction

The following conventions have been used when defining the TTCN.GR table proformas and the TTCN.MP grammar.

A.7.2 TTCN.GR Table Proformas

The TTCN.GR notation is defined using a number of different types of table. The following conventions will be used in the description of proformas for these tables:

- a) Bold text (**like this**) shall appear verbatim in each actual table in a TTCN test case;
- b) Text in italics (*like this*) shall not appear verbatim in a TTCN test case. This font is used to indicate that actual text must be substitute for the italicized symbol. Syntax requirements for the actual text can be found in the corresponding TTCN.MP BNF production.

EXAMPLE 1 — *SuiteIdentifier* corresponds to production 3 in {Annex A}

A.7.3 Syntactic Metanotation

Table A.1 defines the metanotation used to specify the extended form of BNF grammar for TTCN (henceforth called BNF):

Table A.1 — The TTCN.MP Syntactic Metanotation

::=	is defined to be
	alternative
[abc]	0 or 1 instances of abc
{abc}	0 or more instances of abc
{abc} +	1 or more instances of abc
(...)	textual grouping
abc	the non-terminal symbol abc
\$abc	the terminal symbol \$abc
"abc"	the terminal symbol abc

A.7.4 TTCN.MP Syntax Definitions

A.7.4.1 Complete tables defined in TTCN.GR are represented in TTCN.MP by productions of the kind:

\$BEGIN_KEYWORD \$END_KEYWORD

EXAMPLE 2 — $TS_PARdcls ::= \$Begin_TS_PARdcls \{TS_PARdcl\} + \$End_TS_PARdcls$

Normally, these productions contain at least one mandatory field.

A.7.4.2 Lines of a table, i.e. sets of fields in a table, are represented by productions of the kind:

\$KEYWORD \$END_KEYWORD

BEGIN does not appear in the opening keyword.

EXAMPLE 3 — $TS_PARdcl ::= \$TS_PARdcl \ TS_PARid \ TS_PARtype \ PICS_PIXIT \ [Comment] \ \End_TS_PARdcl

A.7.4.3 Individual fields in a line are represented by:

\$KEYWORD

There is no closing keyword.

EXAMPLE 4 — $TS_PARid ::= \$TS_PARid \ TS_PARidentifier$

NOTE Symbols such as TS_PARid can only be used to name a field. The contents of the field must be called, in that case, $TS_PARidentifier$, further defined as an identifier.

EXAMPLE 5 — $TS_PARidentifier ::= Identifier$

A.7.4.4 Sets of tables, up to and including the test suite, are represented by productions of the kind:

\$KEYWORD \$END_KEYWORD

EXAMPLE 6 — $ASPdcls ::= \$ASPdcls \ [TTCN_ASPdcls] \ [ASN1_ASPdcls] \ \$End_ASPdcls$

EXAMPLE 7 — $Suite ::= \$Suite \ SuiteId \ SuiteOverview \ Declarations \ DynamicPart \ ConstraintsPart \ \$End_Suite.$

A.7.4.5 All other productions defining non-terminal symbols have no keywords at the beginning or the end of the right-hand expression.

EXAMPLE 8 — $TimerIdentifier ::= Identifier$

Most of the terminal symbols used in the TTCN.MP grammar are defined in clause {A.3.9}.

A.7.5 TTCN.MP and TTCN.GR Symbols

a) TTCN keywords (terminal symbols) that belong only to the TTCN.MP form start with the dollar character (\$):

EXAMPLE 9 — $\$SuiteId$

b) TTCN keywords (terminal symbols) that belong to both the TTCN.MP and the TTCN.GR forms do not start with the dollar character (\$):

EXAMPLE 10 — the TTCN keyword REPEAT

A.7.6 Uniqueness of Names in TTCN Test Suites

A.7.6.1 Identifiers used within TTCN test suites are case sensitive. Whenever in the TTCN an identifier used from a protocol standard* contains “-” (dash) the dash shall be replaced by “_” (underscore).

A.7.6.2 All identifier names of the following items shall have unique meaning throughout the test suite (i.e. declarations and constraints).

- a) Predefined TTCN types;
- b) user defined TTCN types;
- c) User defined operators;
- d) Test suite parameters;
- e) Test suite constants;
- f) Test suite variables;
- g) Test case variables;

- h) PCO types;
- i) PCO names;
- j) Timer identifiers;
- k) Abbreviation names;
- l) ASP types;
- m) Parameter group types;
- n) PDU types;
- o) Field group types;
- p) ASP constraint names;
- q) Parameter group constraint names;
- r) PDU constraint names;
- s) Field group constraint names;
- t) Test case references;
- u) Test case identifiers;
- v) Test step references;
- w) Test step identifiers;
- x) Default references;
- y) Default identifiers.

A.7.6.3 When ASN.1 is used in a TTCN test suite, ASN.1 identifiers from the following list shall be unique throughout the test suite, regardless of whether the ASN.1 definition is explicit or implicit by reference:

- a) identifiers occurring in an ASN.1 ENUMERATED type as distinguished values;
- b) identifiers occurring in a “NamedNumberList” of an ASN.1 INTEGER type;
- c) “UserTypeIdentifiers” of a User ASN.1 Type Definition.

A.7.6.4 The names of ASP parameters shall be unique within the ASP in which they are declared. The names of PDU fields shall be unique within the PDU in which they are declared.

A.7.6.5 The names of parameters within a parameter group shall be unique within each ASP where it will expand. The names of fields within a field group shall be unique within each PDU (and ASP, if applicable) where it will expand.

A.7.6.6 Labels used within a tree shall be unique within a tree.

A.7.6.7 The tree header names used for local test steps shall be unique within the dynamic behaviour description in which they appear.

A.7.6.8 The formal parameter names which may optionally appear as part of the following shall be unique within that formal parameter list:

- a) User defined operations declaration;
- b) Tree header of a local tree;
- c) Test step identifier;
- d) Defaults identifier;
- e) Parameterized constraint declaration.

If usage of a formal parameter name within a dynamic behaviour description clashes with any other identifier defined in the test suite, the formal parameter name shall take precedence.

A.8 TTCN Test Suite Structure

A.8.1 Introduction

TTCN allows a test suite to be hierarchically structured in accordance with clause 8.1 of Part 1 of this multipart standard*. The components of this structure are:

- a) test groups;
- b) test cases;
- c) test steps;

d) test events.

A TTCN test suite may be completely flat (i.e. have no structure) in which case there are no test groups. TTCN allows the use of test step groups and default groups, similar to the concept of test groups, in order to hierarchically structure test steps and defaults. (This hierarchical structure is optional).

A.8.2 Test Group References and Test Case References

A.8.2.1 TTCN supports a naming structure that shows a conceptual grouping of test cases. Test groups can be nested. Test cases can also be stand alone (see ISO 9646-1, clause 8, Figure 10): references to TTCN test groups shall have the following syntax:

SYNTAX DEFINITION:

• TestGroupReference ::= SuiteIdentifier "/" {TestGroupIdentifier "/" }+

EXAMPLE 11 — A Transport group reference: TRANSPORT/CLASS0/CONN_ESTAB/

A.8.2.2 References to the test cases shall have the following syntax:

SYNTAX DEFINITION:

• TestCaseReference ::= (TestGroupReference TestCaseName) | (SuiteIdentifier "/" TestCaseName)

EXAMPLE 12 — Transport Test Case References:

TRANSPORT/INIT

TRANSPORT/CLASS0/CONN_ESTAB/LT_INIT

where TRANSPORT is the name of the test suite, CLASS0 and CONN_ESTAB are test groups, and INIT and LT_INIT are test case names

A.8.2.3 The test case references define the structure of the test suite.

A.8.3 Test Step Group References and Test Step References

A.8.3.1 Decomposition of a test case (or test step) into test steps is achieved by attaching behaviour sub-trees using the TTCN ATTACH statement. In TTCN the terms test step and sub-tree are synonymous. The position of a test step in the test suite structure is not explicit in the test step reference. The concept of a test step may be expressed in several ways in TTCN. A test step may be:

- implicit in the behaviour tree;
- local to a test case or test step;
- globally accessible as a member of the test step library.

A.8.3.2 In the case of an implicit test step it is impossible to identify or attach the test step. A local test step has no test step reference — it is identifiable through a local tree name. In the case of a global test step, the test step reference specifies the test step's location in the test step library. The test step library has no influence on the test suite structure itself. Test step groups and test step references shall have the following syntax:

SYNTAX DEFINITION:

• TestStepGroupReference ::= SuiteIdentifier "/" {TestStepGroupIdentifier "/" }+

• TestStepReference ::= (TestStepGroupReference TestStepName) | (SuiteIdentifier "/" TestStepName)

EXAMPLE 13 — Transport Test step references:

TRANSPORT/STEP_A

TRANSPORT/STEP_LIBRARY/CLASS0/CONN_ESTAB/STEP_B

A.8.4 Default Group References and Default Behaviour References

Default behaviours (if any) are located in a default behaviours library.

A default reference specifies the default's location in the default library. The defaults library has no influence on the test suite structure itself. Default groups and default references shall have the following syntax:

SYNTAX DEFINITION:

• DefaultGroupReference ::= SuiteIdentifier "/" {DefaultGroupIdentifier "/" }

• DefaultReference ::= (DefaultGroupReference DefaultName) | (SuiteIdentifier "/" DefaultName)

EXAMPLE 14 — Transport default references:

TRANSPORT/DEF_A
TRANSPORT/DEFAULT_LIBRARY/CLASS0/DEF_B

A.9 Components of a TTCN Test Suite

An abstract test suite written in TTCN shall have the following four sections in the following order:

- a) Suite Overview (clause A.10),
which is the information needed for the general presentation and understanding of the test suite, such as test references and a description of its overall purpose;
- b) Declarations Part (clause A.11),
which is the set of components that comprise the test suite (e.g. PCOs, Timers, ASPs, PDUs, and their parameters or fields) is described. This section shall contain the definition of any abbreviations to be used later in the test suite;
- c) Constraints Part (clauses A.12, A.13, A.14),
which is the set of values for the ASPs, PDUs, and their parameters used in the Dynamic Part. The constraints shall be specified using:
 - 1) TTCN tables; or
 - 2) the ASN.1 Modular Method; or
 - 3) both TTCN tables and the ASN.1 Modular Method.
- d) Dynamic Part (clause A.15),
which comprises three sections that contain tables specifying test behaviour expressed mainly in terms of the occurrence of ASPs at PCOs. These sections are:
 - 1) the test case dynamic behaviour descriptions;
 - 2) a library containing test step dynamic behaviour descriptions (if any);
 - 3) a library containing default dynamic behaviour descriptions (if any).

SYNTAX DEFINITION:

- Suite ::= \$Suite SuiteId SuiteOverview Declarations ConstraintsPart DynamicPart \$End_Suite

A.10 Suite Overview

This section shall include at least the following information:

- a) the name of the test suite;
- b) references to the relevant base standards;
- c) a reference to the PICS proforma;
- d) a reference to the partial PIXIT proforma (see ISO 9646-2 clause 14);
- e) a reference to where in the Abstract Test Suite specification the mapping of the PICS and PIXIT entries used in test case selection is specified;
- f) an indication of the test method or methods to which the test suite applies, plus for the Coordinated Methods a reference to where the Test Management Protocol is specified. When TTCN is being used for an abstract test suite then there shall only be one test method, but when it is being used for a generic test suite then it may be that different test cases will be written in the style of different test methods;
- g) other information which may aid understanding of the test suite, such as how it has been derived; this should be included as a comment;
- h) a three-part test suite index:
 - 1) a test case index, consisting of the test case identifier, test case reference, page number and a description of each test case (shortened form of the test purpose). The test cases shall be organized according to the structure of the test suite, optionally giving test group identifier or test group reference, page and test group objective (in the description column) where appropriate in the structure to show the test groups;

NOTE There is no need to include test group identifiers in the first column because they are included in the test group references

2) a test step index, consisting of the test step identifier, test step reference, page number and a description of each test step (shortened form of the test step objective). The test steps shall be organized according to the structure of the test step library;

3) a defaults index, consisting of the defaults identifier, defaults reference, page number and a description of each default (shortened form of the default objective). The defaults shall be organized according to the structure of the defaults library.

This information shall be provided in the format shown in the following proforma:

Test Suite Overview			
Suite Name:		<i>SuiteIdentifier</i>	
Standards ref:		<i>Reference</i>	
PICS proforma ref:		<i>Reference</i>	
PIXIT proforma ref:		<i>Reference</i>	
PICS/PIXIT use:		<i>Reference</i>	
Test Method(s):		<i>FreeText</i>	
Comments:		<i>FreeText</i>	
Test Group or Case Identifier	Test Group or Case Reference	Page	Description
. . <i>TestCaseIdentifier</i> <i>TestGroupIdentifier</i> .	. . <i>TestCaseReference</i> <i>TestGroupReference</i> .	. . <i>Number</i> . .	. . <i>FreeText</i> . .
Test Step Identifier	Test Step Reference	Page	Description
. . <i>TestStepIdentifier</i> . .	. . <i>TestStepReference</i> . .	. . <i>Number</i> . .	. . <i>FreeText</i> . .
Default Identifier	Default Reference	Page	Description
. . <i>DefaultIdentifier</i> . .	. . <i>DefaultReference</i> . .	. . <i>Number</i> . .	. . <i>FreeText</i> . .

Proforma 1: Test Suite Overview

SYNTAX DEFINITION:

- SuiteOverview ::= **\$Begin_SuiteOverview** SuiteId StandardsRef PICSref PIXITref HowUsed TestMethods [Comment] SuiteIndex **\$End_SuiteOverview**
- SuiteIdentifier ::= Identifier
- Reference ::= BoundedFreeText
- FreeText ::= {ExtendedAlphaNum}
- SuiteIndex ::= **\$SuiteIndex** TestCaseIndex TestStepIndex DefaultIndex **\$End_SuiteIndex**
- TestCaseIndex ::= **\$TestCaseIndex** {[(TestCaseId [TestCaseRef]) | ([TestGroupId] [TestGroupRef])] Description} + **\$End_TestCaseIndex**
- TestGroupRef ::= **\$TestGroupRef** TestGroupReference
- TestStepIndex ::= **\$TestStepIndex** {TestStepID TestStepRef Description} **\$End_TestStepIndex**
- DefaultIndex ::= **\$DefaultIndex** {DefaultID DefaultRef Description} **\$End_DefaultIndex**

If a dollar symbol (\$) is needed in free text then it shall be preceded by the special character backslash (\), otherwise Free Text shall not include a dollar symbol (\$). Free Text shall not end with the character backslash.

A.11 Declarations

A.11.1 Introduction

The purpose of the declarations section is to describe all the components used in the test suite. All components referenced in the dynamic part shall have been declared in the declarations part. These components are:

- a) User defined types (clause A.11.2.3);
- b) User defined operations (clause A.11.4.3);
- c) Test suite parameters (clause A.11.5);
- d) Test suite constants (clause A.11.6);
- e) Test suite variables (clause A.11.7.1);
- f) Test case variables (clause A.11.7.2);
- g) PCO declarations (clause A.11.8);
- h) Timer declarations (clause A.11.9);
- i) Abbreviations (clause A.11.10).
- j) ASP type declarations (clause A.11.11);
- k) ASP parameter group type declarations (clause A.11.11.3);
- l) PDU type declarations (clause A.11.12);
- m) PDU field group type declarations (clause A.11.12.3);

SYNTAX DEFINITION:

• Declarations ::= **\$Declarations** [UserTYPEdefs] [UserOPdefs] [TS_PARdcls] [TS_CONSTdcls] [TS_VARdcls] [TC_VARdcls] PCOdcls [TIMERdcls] [Abbreviations] [ASPdcls] PDUdcls
\$End_Declarations

A.11.2 General TTCN Types

A.11.2.1 Introduction

TTCN supports both a number of predefined types and mechanisms that allow the definition of user declared types. These types may be used throughout the test suite and may be referenced when test suite parameters, constants, variables, ASP parameters or PDU fields are defined.

SYNTAX DEFINITION:

• Type ::= PredefinedType | UserDefinedType

A.11.2.2 Predefined TTCN Types

A number of commonly used types are predefined for use in TTCN. These types may be referenced even though they do not appear in a type declaration in a test suite. All other types used in a test suite shall be declared in the User Type Declarations and referenced by name.

- a) **INTEGER Predefined Type:** a type with distinguished values which are the positive and negative whole numbers, including zero;
- b) **BOOLEAN Predefined Type:** a type consisting of two distinguished values;
- c) **BITSTRING Predefined Type:** a type whose distinguished values are the ordered sequences of zero, one, or more bits;
- d) **HEXSTRING Predefined Type:** a type whose distinguished values are the ordered sequences of zero, one, or more semi-octets, a semi-octet being an ordered sequence of four bits;
- e) **OCTETSTRING Predefined Type:** a type whose distinguished values are the ordered sequences of zero, one, or more octets, each octet being an ordered sequence of eight bits; thus an OctetString is an even number of HEXSTRING digits;
- f) **Character String Predefined Types:** types whose distinguished values are zero, one, or more characters from some character set; the character string types listed in Table A.2 may be used; they are defined in section 2 of ISO 8824.

Table A.2 — Predefined Character String Types

<i>NumericString</i>
<i>PrintableString</i>
<i>TeletexString</i> (i.e. T61String)
<i>VideotexString</i>
<i>VisibleString</i> (i.e. ISO646String)
<i>IA5String</i>
<i>GraphicString</i>
<i>GeneralString</i>

SYNTAX DEFINITION:

- PredefinedType ::= **INTEGER** | **BOOLEAN** | PredefinedStringType
- PredefinedStringType ::= **BITSTRING** | **HEXSTRING** | **OCTETSTRING** | CharacterString
- CharacterString ::= **NumericString** | **PrintableString** | **TeletexString** | **VideotexString** | **VisibleString** | **IA5String** | **GraphicString** | **GeneralString**

A.11.2.3 User Defined Types for a Specific Test Suite**A.11.2.3.1 Introduction**

Types specific to a test suite may be introduced by the TTCN user. These may be defined using the TTCN user type definitions table and/or ASN.1.

SYNTAX DEFINITION:

- UserTYPEdefs ::= **\$UserTYPEdefs** [TTCN_TYPEdefs] {ASN1_TYPEdef} **\$End_UserTYPEdefs**

A.11.2.3.2 User Type Definitions — Tabular Form

To define a new type, the following information shall be provided:

- a name for the type;
- the base type (if any),

which is required when defining a new type equivalent to or a subset of a previously defined, or predefined, type (see c2A, c2B); the test case writer shall ensure that the base type is compatible with the new type being defined;

the base type field may be left blank when the definition of the base type is implicit in the definition of a restricted length type (see c2B);

- a definition of the type, provided in one of the following manners:

- 1) by listing the set of distinguished values of another type; these values comprise the new type;
- 2) by specifying a subset of the distinguished values of another type; this may be done in a number of ways:

A) by specifying a subrange within a list of values, or integer;

B) by restricting the length of a predefined or user defined *string* type (e.g. BITSTRING, HEXSTRING, IA5STRING).

The length shall be specified in the unit represented by the corresponding string type:

BITSTRING in number of bits, HEXSTRING in number of hexadecimal digits, OCTETSTRING in number of octets and other character strings in number of characters.

This information shall be provided in the format shown in the following proforma:

User Type Definitions			
Name	Base Type	Definition	Comments
.	.	.	.
<i>UserTypeIdentifier</i>	<i>BaseType</i>	<i>TypeDefinition</i>	<i>Free Text</i>
.	.	.	.
.	.	.	.

Proforma 2: User Type Definitions

SYNTAX DEFINITION:

- **TTCN_TYPEdefs** ::= **\$Begin_TTCN_TYPEdefs** {TTCN_TYPEDef} + **\$End_TTCN_TYPEdefs**
- **TTCN_TYPEDef** ::= **\$TTCN_TYPEDef** UserTypeId [Base] TypeDef [Comment]
- **\$End_TTCN_TYPEDef**
- **UserTypeId** ::= **\$UserTypeId** UserTypeIdentifier
- **UserTypeIdentifier** ::= Identifier
- **Base** ::= **\$Base** Type
- **FIELDgroupConstraintTypeDef** ::= **\$TypeDef** TypeDefinition
- **TypeDefinition** ::= TypeIfNoBaseUsed | TypeIfBaseUsed
- **TypeIfNoBaseUsed** ::= TypeAndLength
- **TypeIfBaseUsed** ::= Range | VALlist
- **Range** ::= "(" SignedNumber To SignedNumber ")"
- **To** ::= **TO** | ".."
- **VALlist** ::= "(" SignedValue {Comma SignedValue} ")"
- **Comma** ::= ","
- **TypeAndLength** ::= StringType "[" Number "]"
- **SignedNumber** ::= ["-"] Number
- **SignedValue** ::= ["-"] Value

Where a range of values is used as a TTCN type definition, that range shall be stated with the lower of the two values on the left.

User Type Definitions			
Name	Base Type	Definition	Comments
Transport_Classes	INTEGER	(0, 1, 2, 3, 4)	Classes that may be used for Transport layer connection
Class_Number	INTEGER	(0 .. 4)	

Example 15: Example User Type Definition

A.11.2.3.3 User Type Definitions in ASN.1

User types specified in ASN.1 (ISO 8824) shall be defined in the following proforma:

User ASN.1 Type Definition	
Type Name:	<i>UserTypeIdIdentifier</i>
ASN.1 Definition or Reference	
<div style="text-align: center;"> . . <i>ASN1Ref</i> <i>ASN1Def</i> . . . </div>	

Proforma 3: User ASN.1 Type Definition

SYNTAX DEFINITION:

- *ASN1_TYPEdef* ::= **\$Begin_ASN1_TYPEdef** UserTypeId *ASN1RefOrDef* **\$End_ASN1_TYPEdef**
- *ASN1RefOrDef* ::= **\$ASN1RefOrDef** (*ASN1Ref* | *ASN1Def*) **\$End_ASN1RefOrDef**
- *ASN1Ref* ::= **#REF** {SymbolsFromModule}+
- SymbolsFromModule ::= SymbolList **FROM** ModuleReference
- SymbolList ::= TypeReference {Comma TypeReference}
- TypeReference ::= /* Defined in ISO 8824 */
- ModuleReference ::= /* Defined in ISO 8824 */
- *ASN1Def* ::= (TypeAssignment) + /* Defined in ISO 8824 */

When an ASN.1 reference is used the free text entry shall be preceded by the keyword **#REF**. This applies throughout TTCN where ASN.1 references are used.

A.11.2.4 Type Compatibility for ASN.1 Types

Data objects are assignment compatible:

- a) if their type names are identical;
- b) or if one type is derived from the other type by:
 - 1) subtyping (i.e. subrange types, length restriction);
 - 2) or renaming the original type;

EXAMPLE 16 — *int* ::= **INTEGER**

Data objects of type *int* and **INTEGER** are assignment compatible.

- 3) or changing the tag of the original type;

EXAMPLE 17 — *int2* ::= [5] *int*

int and *int2* are assignment compatible.

Constants that are local to the type definition may be defined using ASN.1 (Named Values) are only compatible to the type they were defined in.

If type A is assignment compatible with type B, and type B is assignment compatible with type C, then type A is always assignment compatible with type C (i.e. assignment compatibility is transitive).

A.11.2.5 Type Compatibility for TTCN Types

For types (user defined types, ASP types, PDU Types) defined using the tabular method the rules a) and b1) from clause A.11.2.4 apply.

NOTE Renaming cannot be achieved in the tabular method.

A.11.2.6 Relationship Between ASN.1 and TTCN Types

The following TTCN predefined types are assignment compatible with their corresponding ASN.1 types:

- a) **INTEGER**;
- b) **BOOLEAN**;
- c) **BITSTRING**;