The drawings contained in this Recommendation have been done in Autocad

Recommendation X.403

MESSAGE HANDLING SYSTEMS:

CONFORMANCE TESTING

(Melbourne, 1988)

The CCITT,

considering

(a) the need for Message Handling Systems;

(b) the need to ensure the interoperability of Message Handling Systems;

(c) the need for conformance testing specifications for Message Handling Systems;

(d) that the X.400ÑSeries Recommendations specify Message Handling Systems;

(e) the stateÑofÑtheÑart of OSI testing methodology and notation within CCITTÑISO,

unanimously declares

(1) that this Recommendation describes the testing methodology for Message Handling Systems;

(2) that this Recommendation describes a notation used to define test specifications for Message Handling Systems;

(3) that this Recommendation describes the scope and content of CCITT Conformance Testing Specification Manuals for Message Handling Systems.

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

4 *Abbreviations*

5 *Conventions*

6 *Overview*

7 *Conformance requirements*

8 *Testing methodology*

9 *Structure of test suites*

10 *Information to be supplied by implementors*

11 *Test notation*

12 *Conformance assessment procedures*

Annex A Ñ Test notation

Annex B Ñ IPMS (P2) PICS proformas

Annex C Ñ MTS (P1) PICS proformas

Annex D Ñ RTS PICS proformas

0 Introduction

This Recommendation describes the test methods, test criteria and test notation to be used for the conformance testing of message handling systems based on the 1984 X.400Ñseries of Recommendations as supplemented by the X.400Ñseries Implementor's Guide (version 5).

1 Scope and field of application

The message handling protocols in the scope of this Recommendation are contained in the 1984 X.400Ñseries of Recommendations together with the X.400Ñseries Implementor's Guide (version 5).

Abstract test specifications for these are contained in the CCITT Conformance Testing Specification Manuals associated with this Recommendation:

- Ñ Conformance Testing Specification Manual for IPMS (P2)
- Ñ Conformance Testing Specification Manual for MTS (P1)
- Ñ Conformance Testing Specification Manual for RTS.

Even though these Manuals are referred to by this Recommendation they are not part of it.

While the complete and correct operation of session, transport and other lowerÑlayer protocols is required for interworking the testing of these layers is not in the scope of this Recommendation. On the other hand, X.400 conformance tests should verify that the Reliable Transfer Server (RTS) correctly uses the layers beneath it.

The tests defined in this document apply to interÑdomain working (ADMD to ADMD and ADMD to PRMD). They relate to any MTA or UA in a domain that supports communications with other domains.

Conformance testing of the semantics and syntax of the actual body part information carried in a BODY PART is beyond the scope of this document.

The purpose of this Recommendation is to minimize the time and expense that manufacturers of X.400 implementations and providers of X.400 services must incur to ensure a high degree of interoperability of their equipment. This purpose is achieved by having a set of X.400 conformance test specifications. The successful joint execution of the test specifications by two implementations can be accepted as compelling evidence of the complete and correct operation of these implementations.

The scope and intention of this Recommendation is different from other CCITT Recommendations which define communication services and protocols such as the 1984 X.400Ñseries of Recommendations. The purpose of the latter Recommendations is to unambiguously define a system. However a Recommendation for conformance testing provides a well chosen subset of tests of the virtually infinite number of tests needed to guarantee full compliance to a protocol standard. The subset is chosen in such a way that it gives a high level of confidence that tested implementations will interwork while taking into account pragmatic considerations such as time taken to perform the tests.

Testing for conformance to functional standards is beyond the scope of this Recommendation. However it is recognized that conformance tests for functional standards can be derived from this Recommendation and the associated Test Specification Manuals.

It should be recognized that the conformance testing of message handling systems may fall within the framework of national regulations and may be subject to the testing policies of Administrations which are beyond the scope of this document.

2 References (1984 version)

Recommendation X.210 Open Systems Interconnection (OSI) Layer Service Definitions Convention.

Recommendation X.400 Message Handling Systems: System ModelÑService Elements.

Recommendation X.401 Message Handling Systems: Basic service elements and optional user facilities.

Recommendation X.408 Message Handling Systems: Encoded information type conversion rules.

Recommendation X.409 Message Handling Systems: Presentation transfer syntax and notation.

Recommendation X.410 Message Handling Systems: Remote operations and reliable transfer server.

Recommendation X.411 Message Handling Systems: Message transfer layer.

Recommendation X.420 Message Handling Systems: Interpersonal messaging user agent layer.

X.400 Series Implementor's Guide version 5.

3 Definitions

3.1 *Service convention definitions*

This Recommendation makes use of the following terms defined in Recommendation X.210, (version 1984):

a) primitive;

b) request (primitive);

c) indication(primitive);

d) response (primitive);

e) confirm (primitive).

3.2 *Message handling definitions*

This Recommendation makes use of the following terms defined in Recommendation X.400, (version 1984):

a) administration management domain;

b) interpersonal message (Recommendation X.420);

c) message;

d) message transfer (Recommendation X.411);

e) originator;

f) private management domain;

g) recipient;

h) user.

4 Abbreviations

The following abbreviations are used in this Recommendation:

ADMD Administration Management Domain

ASP Abstract Service Primitive

DSE Distributed Single layer Embedded testmethod

MHS Message Handling System

IPMS Interpersonal Messaging System

IUT Implementation Under Test

MPDU Message Protocol Data Unit

MT Message Transfer

MTA Message Transfer Agent

MTS Message Transfer System

P1 The Message Transfer Protocol [X.411]

P2 The Interpersonal Messaging Protocol [X.420]

PCO Point of Control and Observation

PICS Protocol Implementation Conformance Statement

PIXIT Protocol Implementation Extra Information for Testing

PDU Protocol data unit

PRMD Private management domain

RTS Reliable Transfer Server

SAP Service Access Point

TSP Test Suite Parameter

TTCN Tree and Tabular Combined Notation

UA User Agent.

5 Conventions

No conventions are defined for this Recommendation.

6 Overview

There are two kinds of CCITT documents concerned with X.400 Conformance testing:

(a) This CCITT Recommendation entitled ÒX.403 Message Handling Systems Ñ Conformance testingÓ;

(b) Three associated CCITT Conformance Testing Specification Manuals entitled:

- Ñ Conformance Testing Specification Manual for IPMS (P2)
- Ñ Conformance Testing Specification Manual for MTS (P1)
- Ñ Conformance Testing Specification Manual for RTS

The CCITT Recommendation is intended for a wide readership. The Manuals are intended for test implementors and contain detailed test specifications.

6.1 *The X.400 conformance testing Recommendation*

This Recommendation gives the following information:

a) Conformance requirements of X.400 implementations.

b) The testing methodology.

c) The structure of the test specifications.

d) Information to be supplied by implementors as a prerequisite to conformance testing.

e) The test notation.

f) Conformance assessment procedures.

6.2 *The X.400 conformance testing specification manuals*

Three CCITT conformance testing specification manuals contain test specifications for the IPMS (P2), MTS (P1), RTS. The test specifications are written in a notation described in general terms in ¤ 11. The conformance testing specification manuals are referred to by this Recommendation but they are not part of it.

Since the manuals contain detailed and unambiguous test specifications, users of these manuals should be familiar with the X.400Ñseries of Recommendations and with the testing methodology used.

7 Conformance requirements

The purpose of the test specifications referenced by this Recommendation is to define tests that will establish to a high degree of confidence that the various protocol layers of an implementation under test conform to the requirements of the X.400Ñseries of Recommendations (1984).

7.1 A system claiming to conform to the X.400 IPMÑService has to support correctly:

- Ñ the basic IPM service elements as defined in Table 2/X.400;
- Ñ the IPM Optional User facilities defined as Essential in Tables 1/X.401 and 2/X.401 (where the categorization for origination and reception should be considered);
- Ñ the IPM Optional User facilities defined as Additional in Tables 1/X.401 and 2/X.401, which are claimed to be supported;

- Ñ the requirements related to the IPM service as defined in version 5 of the X.400Ñseries Implementor's Guide.

7.2 A system claiming to conform to the X.400 MTÑservice has to support correctly:

- Ñ the basic MT service elements as defined in Table 1/X.400 related to the MTS (P1) protocol;
- Ñ the MT Optional User facilities defined as Essential in Tables 3/X.401 and 4/X.401 and related to the MTS (P1) protocol;
- Ñ the MT Optional User facilities defined as Additional in Tables 3/X.401 and 4/X.401 and related to the MTS (P1) protocol, which are claimed to be supported;
- Ñ the requirements related to the P1 MTÑservice as defined in version 5 of the CCITT X.400Ñseries Implementor's Guide.

7.3 system claiming to conform to the X.400 RTSÑservice has to support correctly:

- Ñ the RTSÑservices as defined in X.410;
- Ñ the requirements related to the RTSÑservice as defined in version 5 of the CCITT X.400Ñseries Implementor's Guide.

7.4 Claims of conformance of an implementation to the X.400Ñseries of Recommendations can be tested using the conformance testing specification manuals associated with this Recommendation to ensure that:

(a) The implementation does not act or react in a way different to the one described in the Recommendations.

(b) The implementation is capable of handling protocol errors.

The reaction of an implementation on receipt of protocol errors is not defined in the X.400Ñseries of Recommendations. For the purpose of conformance testing the minimum additional requirement is made that the implementation subsequently continues to operate normally in such cases.

The absence of a mandatory protocol element in P2 or P1 is regarded as a protocol error. It should be noted that in an implementated MHS a recipient domain may choose to deliver an incorrect MPDU. This should be considered as proprietary design by the equipment vendor, and the specific actions taken in these situations are defined by the vendor and not subject to conformance.

(c) The implementation correctly handles the requirements defined in X.400 Implementor's Guide VersionÊ5.

Maximum lengths and maximum number of occurrences are interpreted in the following way:

- Ñ on origination: the implementation may support maximum lengths/occurrences up to but not exceeding the constraint value.
- Ñ on reception: the implementation must support the maximum lengths/occurrences of the constraints. Values above the constraints may be supported but the conformance requirements on the implementation upon reception of a length/occurrence exceeding the constraint are the same as for protocol errors.

Claims of conformance to the X.400 series of Recommendations can not be tested for those implementations for which it is not possible to perform all the required tests for features labeled mandatory, basic or essential optional.

8 Testing methodology

8.1 *Test configurations*

Two test configurations are used. The first configuration is shown in Figure 1/X.403 and is used to test IPMS (P2), MTS (P1) and RTS.

Fig. 1/X.403/T0704030-88 = 5cm

The second configuration is shown in Figure 2/X.403 and is used to test the relay aspects of the MTS (P1) protocol.

Fig. 2/X.403/T0704040-88 = 5cm

8.2 *Points of control and observation*

Test cases are described abstractly in terms of events at Points of Control and Observation (PCO) in both the tester and the Implementation Under Test (IUT). These PCOs are generally Service Access Points (SAPs) and the events are generally Abstract Service Primitives (ASPs). This does not imply that manufacturers are required to have accessible SAPs or to implement ASPs within their systems. During test execution the PCOs of an IUT may be accessed indirectly through a user interface. Where testing is performed through a user interface, the mapping of events between the SAP and the user interface is provided by the supplier of the IUT as described in ¤ 10.2.

8.2.1 *PCOs for IPMS(P2)*

The IPMS (P2) test cases are described using the Points of Control and Observation (PCOs) shown in FigureÊ3/X.403.

Fig. 3/X.403/T0704050-88 = 5cm

For the tester, the Point of Control and Observation is the Service Access Point (SAP) defined at the boundary between the User Agent Layer and the Message Transfer Layer. This PCO makes use of the Message Transfer Layer Service Primitives defined in Recommendation X.411.

For the IUT, the PCO is the SAP defined at the upper boundary of the User Agent Layer. However Recommendation X.420 does not include definition of Service Primitives and it has therefore been necessary to construct hypothetical ones for sending and receiving IPÑmessages, in order that the test cases can be described in a formal way.

8.2.2 *PCOs for MTS(P1)*

The MTS (P1) test cases are described using the PCOs shown in Figure 4/X.403.

Fig. 4/X.403/T0704060-88 = 6cm

For the tester, the PCO is the SAP defined at the boundary between the MT Layer and the RTS. This PCO makes use of the RTS primitives defined in Recommendation X.410.

For the IUT, the PCO is the SAP defined at the boundary between the UA Layer and the MT Layer. This PCO makes use of the MT Service Primitives defined in Recommendation X.411.

The testing of relay functions requires more than one tester SAP. Similarly the testing of multiple destination delivery requires more than one UA on the IUT.

8.2.3 *PCOs for RTS*

The RTS test cases are described using the PCOs shown in Figure 5/X.403.

For the tester, the PCO is the SAP defined at the boundary between the RTS and the Session Layer. This PCO makes use of the Session Service Primitives defined in Recommendation X.215.

For the IUT, the PCO is the SAP defined at the upper boundary of the User Agent Layer. This PCO makes use of the same hypothetical Service Primitives defined for IPMS (P2) (¤ 8.2.1).

The description of the RTS test cases includes events at a third SAP at the IUT (SAPÑI) between the MT Layer and RTS. The events of this SAP are used only for clarification and it is not used as a PCO.

Fig. 5/X.403/T0704070-88 = 6cm

8.3 *Test design strategy*

The MHS test specifications are designed using the following concepts:

a) A test specification is defined as a test suite composed of a number of test cases as defined in ¤ 11.1.

b) Test cases are defined in terms of:

- Ñ lower layer ASP events at the tester;
- Ñ upper layer ASP events at the IUT.

c) The test cases define the sequencing of these ASP events and the associated parameters, in particular the PDUs.

d) Test cases for valid behaviour specify ASP event sequences and PDUs that are in accordance with the X.400Ñseries of Recommendations.

e) Test cases for invalid behaviour are characterized by:

- Ñ a correct PDU or event initiated by the tester in a protocol state where it is not permitted (an inopportune event); or
- Ñ a correct PDU incorporating an element which is syntactically correct and in range, but conflicts with the negotiated value; or
- Ñ a PDU sent by the tester which is syntactically incorrect (examples are a missing mandatory protocol element, an outÑofÑrange value or an incorrectly encoded length indicator); or
- Ñ for RTS a lower layer ASP event issued by the tester used with parameters that are not allowed or not appropriate (example SPSN in SConnect) by X.400 restrictions.

f) The depth of testing is restricted to a reasonable number of test cases using the following principles:

1) For valid behaviour:
   - Ñ if there is a small number of valid protocol element values, test all of them;
   - Ñ if there is a range of values, test the bounds and a few common values;
   - Ñ if there are no bounds, test an extreme value besides the common ones.
2) For invalid behaviour:
   - Ñ The number of test cases for a particular type of error is reduced to one or just a few common ones.

8.3.1 *Strategy for X.409 testing*

The X.409 test cases defined in the CCITT conformance testing specification manuals associated with this Recommendation are applicable only to X.400 message handling systems. The testing of X.409 is done as part of the MTS (P1), IPMS (P2) and RTS testing. The features tested are the data types defined in X.409, the various forms of length encoding and the use of primitive and constructor data elements. To increase the likelihood that the tests can be performed, the test cases wherever possible have been defined using the protocol elements associated with mandatory service elements.

Two categories of X.409 tests are identified:

Ñ *Decoding tests*

These tests are constructed by identifying X.409 features to be exercised and devising sets of correctly and incorrectly encoded test PDUs containing these features. The tests are performed by transmitting the test PDUs to the IUT and observing the local reaction of the implementation and/or any PDUs returned to the tester.

Ñ *Encoding tests*

These tests are constructed by identifying a set of user several requests that will generate PDUs whose encoding will exercise major X.409 features. The tester must check the validity of the coding of the resulting PDUs generated by the IUT.

The decoding tests allow the X.409 decoding features of an implementation to be fully exercised using valid and invalid test PDUs. Encoding tests only allow the valid behaviour of X.409 encoding to be checked.

8.3.2 *Strategy for IPMS(P2) testing*

Two categories of test are identified:

Ñ IUT as originator;

Ñ IUT as recipient.

With the IUT as originator, for each service element supported by the implementation, tests are performed by:

Ñ invoking the service;

Ñ the tester checking the validity of the resulting PDUs;

Ñ where appropriate the tester returning valid and invalid response PDUs to the originator.

With the IUT as recipient, for each service element, tests are performed by:

Ñ the tester sending valid and invalid PDUs for that service;

Ñ observing the local reaction of the UA;

Ñ checking the validity of any further PDUs generated by the UA.

In order to avoid unnecessary duplication of test cases, IPM service elements which are also MT service elements (for instance Delivery Notification) are listed in the MTS (P1) test suite in conjunction with the corresponding MT service elements, and not in the IPMS (P2) test suite.

It is assumed that the testing of the MT layer is done through a User Agent.

8.3.3 *Strategy for MTS(P1) testing*

When testing the operation of a MTS (P1) implementation five categories of tests are identified.

Ñ IUT as originator;

Ñ IUT as recipient;

Ñ IUT as relay;

Ñ IUT as relay recipient;

Ñ IUT as recipient/originator.

With the IUT as originator, for each service element supported by the implementation, tests are performed by:

Ñ invoking the service;

Ñ checking the validity of the resulting PDUs.

With the IUT as recipient, for each service element supported by the implementation, tests are performed by:

- Ñ the tester sending valid and invalid PDUs for that service;
- Ñ observing the local reaction of the UA;
- Ñ checking the validity of any further PDUs generated by the UA.

With the IUT as relay, for each service element tests are performed by:

- Ñ the tester sending valid and invalid PDUs for relaying;
- Ñ checking the validity of the reaction of the IUT.

With the IUT as a relay recipient, for each service element tests are performed by:

- Ñ sending a set of valid and invalid PDUs destined for more than one recipient. At least one of these recipients is attached to the IUT and a further recipient is attached to a remote MTA such that the IUT has to relay the message;
- Ñ checking the validity of the reaction of the IUT as recipient;
- Ñ checking that the PDUs that are relayed are not corrupted and are modified appropriately.

With the IUT as a recipient/originator, for each service element supported by the implementation, tests are performed by:

- Ñ invoking the IUT to send a message to multiple recipients. At least one recipient will be attached to the IUT itself and a further recipient will be attached to a remote MTA;
- Ñ checking the validity of the reaction of the IUT as recipient;
- Ñ checking the validity of the PDUs transmitted by the IUT.

8.3.4 *Strategy for RTS testing*

The following testing phases are used:

a) *The connection/association establishment and negotiation phase*

The X.410 Recommendation allows different negotiable options and the negotiation phase is tested exhaustively using valid and invalid elements.

b) *The orderly release of the connection/association*

Only a few tests are required to check the correct implementation of the RTS release features.

c) *The data transfer phase with token exchange*

The data transfer tests check:

- Ñ the correct operation of data transfer using the negotiated values;
- Ñ the correct operation of token exchange;
- Ñ the correct confirmation of confirmed services;
- Ñ the correct reaction to invalid (e.g. nonÑnegotiated) elements.

d) *Recovery*

Tests are performed to check that an IUT can perform correct recovery after:

- Ñ user aborts;
- Ñ provider aborts;
- Ñ exception reports;
- Ñ not acknowledged checkpoints.

9 Structure of test suites

The IPMS (P2) and MTS (P1) test suites have a common structure which differs from that of the RTS test suites.

9.1 *Structure of IPMS(P2) and MTS(P1) test suites*

The IPMS (P2) and MTS (P1) test suites consist of five groups of test cases:

---

a) *Initial tests*

The initial tests check mandatory features in a small number of test cases. They have been defined in order to check that the implementation correctly supports the main mandatory features and that it is sensible to continue with full conformance testing.

b) *X.409 tests*

The X.409 tests check the IUT's encoding and decoding of protocol elements. Decoding tests are performed by transmitting test PDUs to the IUT. Encoding tests are performed by checking PDUs received from the IUT.

c) *Protocol element tests*

Protocol element tests identify test purposes for every protocol element in the IPMS (P2)/MTS (P1) protocols. This is important in ensuring a full test coverage for the IPMS (P2)/MTS (P1) protocols. Many of these tests are necessarily performed as part of the service element tests.

d) *Service element tests*

Service element tests check the capability of the IUT to support the service elements in X.400. Some of these tests are carried out in the initial tests and the X.409 tests. Service element tests include both tests for specific service elements and tests for combinations of interdependent service elements.

e) *Additional test*

The additional test group checks features not covered in the other test groups.

As indicated in a) to e) above the number of test cases has been minimized by taking advantage of the fact that the performance of a given test case may cover more than one test purpose. Figure 6/X.403 shows how some of the test purposes identified in a particular test group may actually be achieved by test cases in another group.

Fig. 6/X.403/T0704080-88 = 6cm

9.2 *Structure of RTS test suites*

The RTS test suite is made up of five groups of test cases:

Ñ association establishment tests;

Ñ asociation release tests;

Ñ data transfer tests;

Ñ association recovery tests;

Ñ X.409 tests.

The association establishment tests check the negotiation of the connection elements.

The association release tests check the orderly release of associations.

The data transfer tests check that data is transferred correctly in accordance with the values of the connection elements negotiated during association establishment.

The association recovery tests check that the IUT can recover from breaks in connection both inside and outside activities.

The X.409 tests check the IUT's encoding and decoding of session service user data.

10 Information to be supplied by implementors

10.1 *Protocol implementation conformance statement (PICS)*

The Protocol Implementation Conformance Statement (PICS) is information supplied by an implementor that specifies the protocol features implemented in a Message Handling System.

This information is used during conformance testing:

- to check that the protocol features that have been implemented are consistent with the conformance requirements, in terms of optional and mandatory features, of the X.400Ñseries of Recommendations;
- to select the originator tests to be executed. Recipient and relay tests will be performed to check the behaviour of the system even when it is requested to handle features that it does not implement.

PICS *proformas* for IPMS (P2), MTS (P1) and RTS are shown in Annexes B, C and D. These *proformas* specify the information to be supplied by an implementor concerning:

- the services that are supported for origination, reception and relay functions;
- the protocol features that have been implemented in order to support the services.

The IPMS (P2) PICS explicitly includes the MTS (P1) service elements made available by the IPMS (P2). In order to avoid duplication with the MTS (P1) test suite, tests for such MTS (P1) service elements are not contained in the IPMS (P2) test suite. Where the testing of MTS (P1) is not performed using a UA, MTS (P1) tests may need to be repeated using a UA in order to ensure conformance to the IPMS (P2).

10.2 *Protocol implementation extra information for testing (PIXIT)*

The Protocol Implementation extra Information for Testing (PIXIT) is supplied by an implementor specifying information needed by a tester to execute a test suite.

The IPMS (P2), MTS (P1) and RTS test suites define the behaviour of the implementation in terms of abstract service primitives. In order to invoke and observe this behaviour during test execution the test operator must know how (if at all) these abstract service primitives can be invoked or observed at the real accessible user interface.

The IPMS (P2), MTS (P1) and RTS PIXIT *proformas* will list all the IUT upper layer abstract service primitives used in the test definitions and will ask the implementor to specify how these primitives can be invoked or observed (if at all).

11 Test notation

11.1 *Definitions*

The notation used to define the MHS test specifications makes use of the following definitions:

a) **test suite**

A set of test cases, possibly combined into nested test groups, necessary to perform conformance testing of an implementation.

The test suites do not imply an order of execution.

b) **test group**

A set of related test cases. Test groups may be nested to provide a logical structuring of test cases.

c) **test case**

Specifies the sequences of test events required to achieve the purpose of the test and to assign a verdict ÒpassÓ, ÒfailÓ or ÒinconclusiveÓ.

d) **test event**

An indivisible unit of test specification at the level of abstraction of the specification (e.g. sending or receiving a single PDU).

e) **user**

A userÑinterface process or a computer application which makes use of an MHS.

11.2 *Notation*

The Conformance Test Suites for Message Handling Systems use the Tree and Tabular Combined Notation as described in Annexe A of this Recommendation.

Each test suite specification is defined in six sections:

1) *Introduction*

This contains an overview describing the scope of the tests and the structure of the test suite.

2) *Summary of test cases*

This is a list of all tests giving the test identifier, the test reference and a short title for each test case in the test suite.

3) *Declarations part*

Declares the names and types of all items to be used in defining the test cases.

4) *Dynamic part*

This is the main body of the test suite and defines test cases in terms of trees of behaviour.

5) *Constraints part*

Specifies the values of the ASPs and PDUs used in the dynamic part.

6) *Cross references*

Provides an index to all values used in the main body of the test suite.

12 Conformance assessment procedures (see Figure 7/X.403)

This Recommendation deals only with abstract test specifications for Message Handling Systems. It does not deal with the realization of these test specifications nor with their execution. This clause in the Recommendation is purely for information purposes to describe in general terms how real testing may be done.

12.1 *Overview of the procedure*

The procedures needed to assess the conformance of an implementation include:

- the completion of the PICS and PIXIT proformas by the supplier of the implementation;
- the assessment of these documents;
- the selection and execution of test cases;
- the analysis of the results and the production of test reports.

12.2 *Analysis of PICS*

The first phase in conformance assessment is to ensure that the features claimed to be supported by an IUT comply with appropriate conformance requirements. The conformance requirements for IPMS (P2), MTS (P1) and RTS implementations are defined in ¤ 7 of this document. This check is performed by analysing the information in the PICS documents.

12.3 *Test case selection*

The tests to be performed are selected primarily on the basis of information in the PICS. For every supported feature claimed in the PICS the corresponding test cases in the test suites are selected and executed to check the correct implementation of these features under an extensive range of valid and invalid conditions.

For nonÑsupported features, some recipient test cases shall be executed to explore the response of the IUT. Since in general the X.400 (1984) Series of Recommendations do not define the expected behaviour in these situations, these tests can be ÒpassedÓ with almost any behaviour apart from catastrophic failure by the IUT.

Information in the PIXIT may also provide some constraints on the test cases that can be executed.

12.4 *Execution of tests*

It is recommended that the testing of Message Handling Systems should be done in the order of RTS, then MTS (P1) and then IPMS (P2) testing.

However the order of test cases in the test suites does not imply an order of execution. Apart from the general recommendation that for IPMS (P2)/MTS (P1) the Initial Test Group should be executed first, the order of execution of tests can be determined by the test operators taking into account their test environment and test tools.

Fig. 7/X.403/T0704090-88 = 10 cm

ANNEX A

(to Recommendation X. 403)

TEST NOTATION

A.1 *Introduction*

This annex is an integral part of this Recommendation and describes the notation used in the test suite manuals.

The test notation described here is based on the test notation called Tree and Tabular Combined Notation (TTCN) that has been developed jointly by ISO and CCITT.

The notation described in this Recommendation is derived from an early form of TTCN and has been developed specifically for use in the MHS conformance testing specifications.

Each of the MHS test suites is specified in five parts:

- Declaration part;
- Dynamic part;
- Constraints part;
- Test case identification;
- CrossÑreferences.

A.2 *Declaration part*

The Declaration Part declares the environment and objects used in the test suites and is composed of 7 sections:

- Test configurations;
- Test suite parameters (TSPs);
- Service access points (SAPs);
- Abstract service primitives (ASPs);
- Protocol data units (PDUs);
- Timers;
- Abbreviations.

A.2.1 *Test configurations*

The points of control and observation are declared in this section.

A.2.2 *Test suite parameters*

Every MHS Test Suite has a set of parameters whose values are fixed prior to testing and which are used to define a specific testing environment.

TSPs are declared in tabular form as shown in Figure AÑ1/X.403.

TEST SUITE PARAMETERS		
NAME	TYPE and RESTRICTIONS	COMMENTS

FIGURE AÑ1/X.403

Test suite parameters

By convention the name of each Test Suite Parameter in the MHS test suites is of the form:

TSP–<name>

A.2.3 *Service access points* (SAPs)

Service Access Points are used as points of control and observation in the MHS Test Suites and are declared in tabular form as shown in Figure AÑ2/X.403.

SAPs	
NAME	ROLE

FIGURE AÑ2/X.403

Service access points

By convention the name of a SAP in the MHS Test Suites is generally one capital letter, such as T, U, V (for tester SAPs) or I, J, K (for IUT SAPs).

A.2.4 *Abstract service primitives*

Each ASP type and its associated parameters used in a test suite is declared in tabular form as shown in Figure AÑ3/X.403.

ASP:	SAP:	COMMENTS:	
NAME	RANGE OF VALUES OR TYPE	COMMENTS	C/M

FIGURE AÑ3/X.403

Abstract service primitives

The name of the ASP is specified in the ÒASPÓ field and is derived from the corresponding name in the X.400Ñseries of Recommendations. The SAP at which the ASP occurs is specified in the ÒSAPÓ field. The parameters of the ASP are declared in the ÒNAMEÓ column together with information in ÒRANGE OF VALUES OR TYPEÓ, ÒCOMMENTSÓ and ÒConditional/MandatoryÓ columns.

Since there are no IPMS (P2) ASPs defined in the Recommendations, in order to describe conformance tests it has been necessary to construct hypothetical ASPs at the upper boundary of the User Agent Layer. This does not imply, however that manufacturers are required to implement these ASPs within their systems. It serves only to formalize the requirements for observation and invocation of IPMS service elements by the use of these new ASPs. The relation between IPMS service elements and the actual behaviour of the IUT has to be specified in the implementationÑdependent PIXIT.

A.2.5 *Protocol data units*

The PDU types used in test suites are declared in the form of tables as shown in Figure AÑ4/X.403. These PDUs are not defined explicitly in the test suite, but are given a precise reference to the full definition in the X.400 Recommendations, in the type name section of the table.

DATA TYPE DECLARATION
COMMENTS:
TYPE NAME:

FIGURE AÑ4/X.403

Protocol data units

A.2.6 *Timers*

This section declares the timers to be used. Timer values are expressions in terms of Test Suite Parameters, and are fixed for the whole test suite. Timer values are declared in tabular form as shown in Figure AÑ5/X.403.

TIMER DECLARATION		
TIMER NAME	VALUE	COMMENT

FIGURE AÑ5/X.403

Timers

A.2.7 *Abbreviations*

Abbreviations used in the Test Suite are defined in the form of a table as shown in Figure AÑ6/X.403.

ABBREVIATIONS		
ABBREVIATION	FULL NAME	COMMENT

FIGURE AÑ6/X.403

Abbreviations

A.3 *Dynamic part*

The Dynamic Part defines the test cases of a test suite in terms of trees of behaviour.

Sections A.3.1 and A.3.2 describe generally how trees of behaviour are defined.

Section A.3.3 describes the content and use of Defaults Library.

Section A.3.4 describes the content and use of Test Step Library.

Section A.3.5 describes how each test case in the main body of a test suite is specified.

A.3.1 *Proforma table for test behaviours* (see Figure AÑ7/X.403)

<title> BEHAVIOUR				
IDENTIFIER: <used only for libraries>				
COMMENTS: <used only for libraries>				
DEFAULTS:				
BEHAVIOUR DESCRIPTION	LABEL	CONSTRAINTS REFERENCE	COMMENTS	RESULTS
Extended Comments: <optional>				

FIGURE AÑ7/X.403

Behaviour description

<title> BEHAVIOUR

Title of the behaviour: DEFAULT for the Default Library; DYNAMIC for the Test Step Library and test cases.

IDENTIFIER

This provides a unique identifier for the behaviour description.

DEFAULTS

This lists the identifiers of default behaviour descriptions which are to be used in conjunction with the Dynamic behaviour shown in the ÒBEHAVIOUR DESCRIPTIONÓ part.

BEHAVIOUR DESCRIPTION

Test behaviour is defined using a tree notation as described in A.3.2.

LABEL

The LABELS column may be used to identify events. Branches between events (i.e. ÒGO TOÓ) are specified by Ò->LabelÓ in the behaviour tree.

CONSTRAINTS REFERENCE

For each ASP event of a behaviour tree line, this column gives the reference to the specific ASP value defined in the Constraints Part.

COMMENTS

This column provides comments which ease understanding of the events. Additional comments may be given in the ÒExtended CommentsÓ area. This column can also be used to identify test PDUs associated with test events.

RESULT

This column indicates which test events generate test verdicts. Values of test verdicts are:

Ñ pass: no misbehaviour of the IUT is detected;

Ñ fail: misbehaviour of the IUT is detected;

Ñ inconclusive: the observed behaviour does not allow the assignment of a pass or fail verdict.

A.3.2 *Tree notation for test behaviours*

Trees of behaviour are defined in terms of events which are generally of the form:

<SAP>!<event>

or of the form

<SAP>?<event>

The <SAP> is the point of control and observation at which the <event> occurs. The SAPs used are those declared in the Declaration Part.

The Ò!Ó symbol indicates that the event is sent from the SAP and Ò?Ó indicates that the event is received at the SAP.

The <event> can be:

Ñ an ASP event;

Ñ a timer event;

Ñ an OTHERWISE pseudoÑevent.

A.3.2.1 *Single ASP events*

If the <event> is an ASP event then the names for the ASPs are those specified in the Declaration Part (the value is specified as a reference in the CONSTRAINTS REFERENCE column).

Example line for an ASP event:

I?DELind

This means that a Deliver Indication is received at the IUT's SAP I.

A.3.2.2 *Single timer events*

If <event> is a timer event then it is of the form:

<operation> <parameters>

The ÒstartÓ operation can take one of two forms:

Start <timer type>

Start (<timer type>, <timer id>)

Where <timer type> is defined in the Declaration Part and has a fixed value associated with it defined in terms of TSPs. The <timer id> allows a name to be attached to an instance of a timer type.

The other operations are:

Ñ Cancel: cancels a running or suspended timer;

Ñ Suspend: suspends a running timer;

Ñ Resume: resumes a suspended timer;

Ñ Timeout: expiration of a running timer;

These operations take one of two forms:

<operation> <timer type>

<operation> <timer id>

where <operation> denotes the operation. When the timer was started using the form ÒStart <timer type>Ó, the form Ò<operation> <timer type> must be used; when the timer was started using the form ÒStart <timer id>Ó, the form Ò<operation> <timer id>Ó must be used.

Example:

I!Start T/IÑtimerÑ1

means that at the IUT's SAP I the T/IÑtimerÑ1 (e.g. for a transmission time for a UAPDU to be transferred from the tester to the IUT's user) is started.

I?Timeout T/IÑtimerÑ1

means that at the IUT's SAP I the timeout of the above timer is received.

A.3.2.3 *Single OTHERWISE events*

If <event> is the OTHERWISE pseudoÑevent, this indicates an unspecified event.

Example:

T?OTHERWISE

Means that at the tester's SAP T an unspecified event is received.

A.3.2.4 *Trees of behaviour*

Trees of behaviour combine events in two ways:

Ñ as sequences of events

Ñ as alternative events

The two combination kinds are distinguished by indented and vertical alignment respectively.

Example of a sequence of events:

I!SUBreq

II?SUBcon

I!T?TRNind

This means that first at the SAP I a Submission Request is sent, then at the same SAP a Submission Confirmation is received, after which a Transfer Indication is received at the tester's SAP T.

Example of alternative events:

T?DELind

T?Timeout I/TÑtimer

This means that at SAP T either a Deliver Indication is received or alternatively the timeout is received there.

To build up a complex behaviour tree, the two kinds of combination can be mixed.

Example:

I!SUBreq
II?SUBcon } alternative events
I!T?TRNind
T?DISind

This means that after sending a Submission Request at I, either a Submission Confirmation is received at I, followed by the receipt of a Transfer Indication at T, or a Disconnect Indication is received at T.

A.3.3 *Defaults library*

General default behaviours which are used by several test cases are defined in the Defaults Library using the format shown in Figure AÑ8/X.403. The name of the default is of the form:

LIB−<name>

or

LIB−<name> [X]

where X is a place holder which is replaced by an actual SAP when applying the default element in a particular Test Case.

Note Ñ Where particular default behaviour applies to a single test case only the behaviour table is associated with that test case and the identifier is not prefixed with ÒLIBÑÓ.

DEFAULT BEHAVIOUR				
DEFAULT IDENTIFIER: LIB –				
COMMENTS:				
DEFAULTS:				
BEHAVIOUR DESCRIPTION	LABEL	CONSTRAINTS REFERENCE	COMMENTS	RESULTS
Extended Comments: <optional>				

FIGURE AÑ8/X.403

Default behaviour

A.3.4 *Test step library* (see Figure AÑ9/X.403)

Where a sequence of test steps is of use in several test cases they can be included in the Test Step Library and given a name of the form:

LIB–<name>

Note Ñ Where a test step applies to a single test case the behaviour table is associated with that test case and the identifier is not prefixed with ÒLIBÑÓ.

A.3.5 *Test case* (see Figure AÑ10/X.403)

Each test case in the main body of the test suite is described in terms of three headings a)Ñc), and a behaviour tree d):

a) *Test reference and test identifier*

These elements give a unique reference and identifier for each test case and are described fully in ¤ A.5.

b) *Summary*

A brief overview of the purpose of the test is provided.

c) *Test description (optional)*

This provides an informal description of the actions and events that should take place during the test and an informal verdict criteria.

d) *Behaviour tree*

Dynamic behaviour is described using the tree notation defined in ¤ A.3.2.

DYNAMIC BEHAVIOUR				
TEST STEP IDENTIFIER: LIB–				
COMMENTS:				
DEFAULTS:				
BEHAVIOUR DESCRIPTION	LABEL	CONSTRAINTS REFERENCE	COMMENTS	RESULTS
Extended Comments: <optional>				

FIGURE AÑ9/X.403

Test step behaviour

DYNAMIC BEHAVIOUR				
DEFAULTS: (see note 1)				
BEHAVIOUR DESCRIPTION	LABEL	CONSTRAINTS REFERENCE	COMMENTS	RESULTS
(see Note 2)				(see Notes 3 and 4)
Extended Comments: <optional>				

Note 1 Ñ In this field all Default Library Identifiers used are inserted. Where necessary, the SAP at which they are applied is also identified. If for example the field contains the entry:

LIBnexpected [T]

it means that the subtree associated with this Default Behaviour is considered to be associated with the SAP T.

Note 2 Ñ Test Step Library behaviour is included in the behaviour tree using the following notation:

+<Test Step Library Identifier>.

Note 3 Ñ The behaviour tree of every Test Case provides the verdicts pass, fail, and where appropriate inconclusive.

Note 4 Ñ When using Default Library elements it is possible that some of the verdict alternatives are ÒhiddenÓ in the Default Library element.

FIGURE AÑ10/X.403

Test case behaviour

A.4 *Constraints part* (see Figure AÑ11/X.403)

The Constraints Part of a Test Suite specifies the values and their encoding of all instances of ASPs, Test PDUs, Base PDUs and Library Components. The Constraints Parts is divided into the following sections:

- Ñ Introduction to Constraints Part;
- Ñ ASP Constraints;
- Ñ Test PDU Constraints;
- Ñ Base PDU Constraints;
- Ñ Components Library.

Fig. A-11/X.403/T0704100-88 = 10 cm

A.4.1 *ASP constraints*

Values of ASPs are defined as specific instances of a generic ASP.

A.4.1.1 *Specification of a ÒGenericÓ ASP*

A generic ASP is defined using the format shown in Figure AÑ12/X.403.

The ÒFIELDSÓ column is used to list all the parameters of the ASP.

The ÒVALUE or REFERENCEÓ column is used to specify a value for each parameter and this can be done in four ways:

a) as a reference which can be a TSP name or a library component name;

b) as an explicit value;

c) as ÒÑÓ to indicate that this parameter may be omitted in specific instances of this ASP;

d) as Ò?Ó to indicate that for ÒrequestÓ ASP's this parameter must have a value defined in a specific instance if it is a component of interest.

<ASP name>	<abbreviated name>	GENERIC
FIELDS	VALUE OR REFERENCE	COMMENT
<field–1>	<TSP instant>	M
<field–2>	<Lib component>	M
<field–3>	?	C
<field–4>	Ñ	not applicable
<field–5>	Ñ	C

FIGURE AÑ12/X.403

Generic ASP specification

A.4.1.2 *Specification of ASP instances*

Specific values of ASPs are defined using the tabular format shown in Figure AÑ13/X.403.

<ASP name>	<generic abbreviated name>	INSTANCES
INSTANCE NAME	MODIFIED PARAMETER	VALUE or REFERENCE
<instance–1>	<field–3>	<test–pdu–1>
<instance–2>	<field–3>	<test–pdu–2>
	<field–6>	<Lib.component>
<instance–3>	<field–3>	<test–pdu–3>
	<field–7>	<TSP instant>
	<field–8>	<given value>

FIGURE AÑ13/X.403

Specific ASP value

The ÒINSTANCE NAMEÓ column is used to identify specific instances of the ASP used in the test suite.

The ÒMODIFIED PARAMETERÓ column identifies, for ÒrequestÓ ASP's those parameters whose values are to be modified from the generic ASP specification, and for ÒnotificationÓ ASP's those parameters whose values are to be checked.

The ÒVALUE or REFERENCEÓ column can contain either specific values or references to library components ASPs or test PDUs.

A.4.2 *Specifying PDU values*

The MHS test suite contains a large number of test PDU values. Each PDU is defined in terms of modifications to one of the small number of ÒbaseÓ PDUs.

For convenience commonly used PDU components are defined in a library and are referenced by test PDUs and base PDUs.

A.4.3 *Base PDUs*

A.4.3.1 *Base PDU specification*

Base PDUs are not themselves used as test PDUs but they serve as a basis from which to derive the test PDUs. Usually only a few Base PDUs need to be specified.

The name of a Base PDU is of the form:

BASE–<PDUtypename>–<number>

Example of a Base PDU:

DESCRIPTION	VALUE or REFERENCE	COMMENT
BASE–IMÑUAPDU–1		
SEQUENCE {		
Heading	[BASE–IM–UAPDU–1–Heading]	
Body	[BASE–IMÑUAPDU–1–Body]	
}		
BASE–IMÑUAPDU–1–Heading		
SET {		
IPMessageID	[L–IPMessageID–20]	
}		
BASE–IMÑUAPDU–1–Body		
SEQUENCE OF {		
BodyPart	[L–BodyPart–20]	
}		

The value or value reference of each element of the structure is specified within square brackets (Ó[Ó and Ò]Ó) under the VALUE or REFERENCE heading.

When specifying the encoding of a PDU for encoding/decoding tests,two additional columns are used to specify the ID Code [ID] and Length Indicator [LI] of each element of the PDU. The format for doing this is shown in the example below.

DESCRIPTION	ID	LI	VALUE or REF	COMMENT
L–Phantasy–2				
SEQUENCE {	[`AO`H]	[LI]		IA5Text
SET {	[`31`H]	[LI]		
repertoire INTEGER	[`80`H]	[LI]	[5]	ia5
}				
IA5String	[`36`H]	[`8106`H]		constructor
	[`04`H]	[`01`H]	[Ò1Ó]	
	[`04`H]	[`02`H]	[Ò23Ó]	
}				

The values of ID and LI can be specified explicitly to allow invalid and various forms of valid codings to be defined. The mnemonic ÒLIÓ is used to indicate that any valid encoding of length is allowed.

A.4.3.2 *Identifying the components to be modified*

A component which is to be replaced in a PDU is identified by a path through the declaration of the PDU. The path is written as a list of elements, each separated from the next by a Ò.Ó. The elements in the list can be labels which appear in a BASE PDU, components which appear in the leftÑhand side of a labeled declaration, or components which appear in the leftÑhand side of the expansion of a library reference in the rightÑhand side of a declaration.

For example, consider the following definitions:

```
Instance–
    SET{
        a                   [value]
        b                   [L–Component–1]
  }
L–Component–1
    SET{
        c                   [value]
        d                   [L–Component–2]
  }
L–Component–2
    SEQUENCE
        e                   [value]
    }
```

Note Ñ L–Component–1 is in the Component Library.

In order to reference ÒaÓ, the path would be instance–1.a.

In order to reference ÒeÓ, the path would be instance–1.b.d.e.

A.4.4 *Test PDUs*

Test PDUs are defined in terms of operations on Base PDU's. These operations refer to Library Components, TSPs or specific values.

There are two kinds of test PDU:

- Ñ PDUs sent by the tester (IUT as recipient)

  By convention the names of these PDUs are of the form

  <PDU name>–x–<number>

  where x is the number of the base PDU from which the test PDU is derived.
- Ñ PDUs received by the tester (IUT as originator)

  By convention the names of these PDUs are of the form

  <PDU name>–0<number>

  where Ò0Ó indicates that these test PDUs are not derived from a base PDU.

A.4.4.1 *Test PDUs sent by the tester*

A test PDU sent by the tester to the IUT is normally constructed from a Base PDU by means of the REPLACE operation.

The specification has the form:

DESCRIPTION	VALUE or REFERENCE	COMMENT
<test PDU to be specified>		
<base PDU to be used> REPLACE		
<base PDU part>		
BY		
<partial ASN.1 tree>	[<value>]	

For the conventions of value assignments see ¤ A.4.6.

Example:

DESCRIPTION	VALUE or REFERENCE	COMMENT
IMÑUAPDUÑ1Ñ18		
BASE–IMÑUAPDU–1		
REPLACE		
BASE–IMÑUAPDU. Heading		
BY		
SET {		
IPMessageID	[L–IPMessageID–7]	Library Components
originator ORDescriptor	[L–ORDescriptor–11]	
}		

To construct invalid components in test PDUs to be sent by the tester, the abstract REDEFINE operation is sometimes used. It is used together with the REPLACE operation in the following form:

DESCRIPTION	VALUE or REFERENCE	COMMENT
<test PDU to be specified>		
REDEFINE		
<Type to be redefined>	:: = <new definition>	
<base PDU to be used>		
REPLACE		
<base PDU part>		
BY		
<partial ASN.1 tree>	[<value>]	

The scope of the newly defined type is restricted to the PD definition containing the REDEFINE operation.

Note Ñ That if the <value> is a reference to an element defined elsewhere (i.e. a TSP or a Library Component), then the new type definition does not affect the referenced element itself but only its usage in the actual PDU.

Example:

IM–UAPDU–1–3		
REDEFINE		
ORName :: = [APPLICATION 1] IMPLICIT SEQUENCE {		
	P1.StandardAttributeList	
	P2.DomainDefinedAttributeList	OPTIONAL
BASE–IM–UAPDU–1		
REPLACE		
BASE–IM–UAPDU–1–Heading		
BY		
SET {		
IPMessageID		[L–IPMessageID–1]
originator ORDescriptor		[L–ORDescriptor–1]
}		

The error to be constructed here is the wrong tag of the ORName type (the correct tag would be [APPLICATION 0]). The scope of the erroneous typeÑdefinition constructed by ÒREDEFINEÓ is restricted to all occurrences of ORName in the definition of IMÑUAPDUÑ1Ñ3. This means that LÑORDescriptorÑ1 is used here with the modified ORName type, whereas the usage of this library component in other PDUs or components remains unaltered.

A.4.4.2 *Test PDUs received by the tester*

For received PDUs normally only a portion of the PDU relates to the purpose of the test.

A component of interest is identified and its value assigned using the techniques described in ¤ A.4.3.

The specification scheme has the following form:

DESCRIPTION	VALUE or REFERENCE	COMMENT
<Test PDU to be specified>		
Partial definition Ñ Components of interest		
<Test PDU part>	[<value>]	

Example:

DESCRIPTION		VALUE or REFERENCE	COMMENT
SRÑUAPDUÑOÑ95			
Partial definition Ñ Components of interest			
SRÑUAPDU.CHOICE.nonÑReceipt			
reason	INTEGER	[1]	autoforward
comments	PrintableString	[Òon holidayÓ]	
returned	IMÑUAPDU	L–IMUAPDU–2]	
SRÑUAPDU.report			
IPMessageID		[L–IPMessageID–15]	

A.4.5 *Component library*

Components of PDUs are defined in the library and are referenced in Base PDU specifications, Test PDU specifications and by other library components.

The name of a Library Component is of the form:

L–<ASN.1 type name>–<number>

and is specified using the techniques described in ¤ A.4.3

Example:

DESCRIPTION	VALUE or REFERENCE	COMMENT
L–Phantasy–1		
SEQUENCE {		
SET {		
SET {		
ContentType		
originator	[2]	p2
Pl.ORName	[TSP–OrName–1]	Test Suite Parameter
original		
SET {BIT STRING}	[{`20H`}]	IA5 Text
DeliveryFlags	[`40H`]	Conversion Prohibited
ThisRecipient		
Pl.ORName	[TSP–ORName–1]	
submission TIME	[TPS–UTCTime–1]	
}		
}		
IMÑUAPDU	[L–IMÑUAPDU–1]	Library Cpt
}		

A.4.6 *Value conventions*

The following conventions are used when defining values or value references for PDU components.

Value references identify components defined either within the Component Library or within the Test Suite Parameters section. CharacterString Values can specified within doubleÑquotes (e.g. ÒabcÓ); Bit String Values are specified within singleÑquotes (e.g. '0A'H or '0001'B; hexadecimal or binary notation); Integer Values are specified as numeric characters (e.g. 2); sets and sequences of values are specified within curly brackets separated by comma (e.g. {ÒabcÓ, ÔOA'H}).

For PDU's sent by the tester:

[?] indicates that the value has no influence on the test and may be anything that is legal according to the relevant service or protocol standard;

[Ñ] indicates that the parameter shall be absent;

[*] indicates that the value is to be inserted by the tester before test execution.

For PDU's received by the tester:

[?] indicates that the tester need make no verification of the value of the parameter;

[Ñ] indicates that the tester shall check that the parameter is absent.

Note Ñ That the Ò?Ó and ÒÑÓ symbols in value assignments of PDU components have got other meanings than Ò?Ó and ÒÑÓ in generic ASP schedules.

A.5 *Test case identification*

Test cases are completely identified using four components:

Ñ a test group identifier;

Ñ a subgroup identifier;

Ñ a validity identifier;

Ñ a test number.

These four components are specified in two equivalent ways:

Ñ as a Test Reference where the four components are textual and descriptive;

Example: OriginalEncodedInfoTypeIndication/Recipient/Valid/2

Ñ as a Test Identifier where the four components are numeric and concise.

Example: 307.2.1.2.

A.5.1 *IPMS(P2) and MTS(P1) identification*

A.5.1.1 *Test Groups*

Number ranges have been allocated for the test groups as shown below:

Initial Tests	001 Ñ 099
X.409 Tests	100 Ñ 199
Protocol Elements Tests	200 Ñ 299
(for frequently occurring Elements)	
X.400	
Service Elements Tests	300 Ñ 399
Additional Tests	400 Ñ 499

A.5.1.2 *Subgroups*

Numeric identifiers have been allocated to the test subgroups as shown below:

Originator	1
Recipient	2
Encode	1

Decode	2
Relay	3
RelayingÑRecipient	4
RelayingÑOriginator	5

A.5.1.3 *Validity identifiers*

Test cases which exercise valid behaviour are distinguished from those which exercise the IUT's reaction to invalid behaviour using the numeric identifiers shown below:

Valid	1
Invalid	2

A.5.1.4 *Test case numbers*

Test cases for a particular group/subgroup/validity are numbered sequentially.

A.5.2 *RTS identification*

A.5.2.1 *Test groups*

Number ranges have been allocated for the test groups as shown below:

Association Establishment	1
Association Release	2
Data Transfer	3
Association Recovery	4
X.409 Tests	5

A.5.2.2 *Subgroups*

Numeric identifiers have been allocated to the RTS subgroups as shown below:

Initiator	1
Responder	2
Sender	1
Receiver	2

A.5.2.3 *Validity identifiers*

Test cases which exercise valid behaviour are distinguished from those which exercise the IUT's reaction to invalid and inopportune behaviour using the numeric identifiers shown below:

Valid	1
Invalid	2
Inopportune	3

A.6 *Cross referencing*

A.6.1 *Cross reference numbering*

The MTS (P1) and IPMS (P2) test suites contain a cross referencing system for the ASPs, test PDUs and library components. The cross referencing appears in the left and right margins of the test suite as shown in FigureÊAÑ14/X.403.

Numbers in the left hand margin of the test suite are in sequential order and are Òplace identifiersÓ. They occur whenever an ASP, test PDU or library component occurs in the test suite.

Whenever an ASP, test PDU or library component occurs, numbers are also placed in the right hand margin. These numbers are forward and backward references to the place identifiers of the other occurrences of the ASP, test PDU or library component.

Where a forward or backward reference can not be found then a dot (Ó.Ó) is printed in the right hand margin. This should not occur in fully defined test suites.

Where a line in the test suite contains more than one ASP, test PDU or library component, the cross references for each item in the line are separated by vertical bars (Ó|Ò) in the right hand margin as shown in FigureÊAÑ15/X.403.

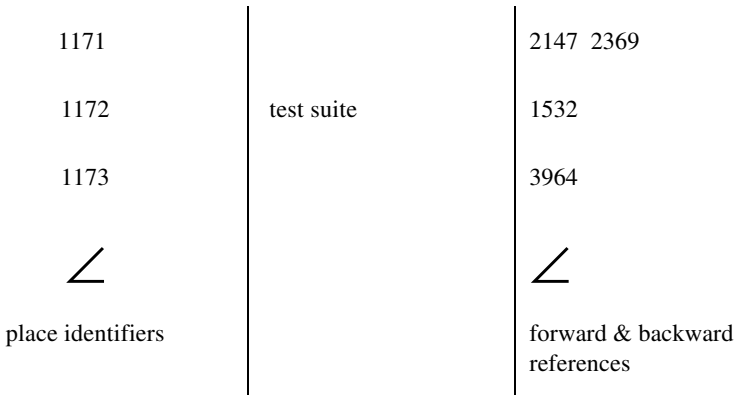

FIGURE AÑ14/X.403

Cross referencing

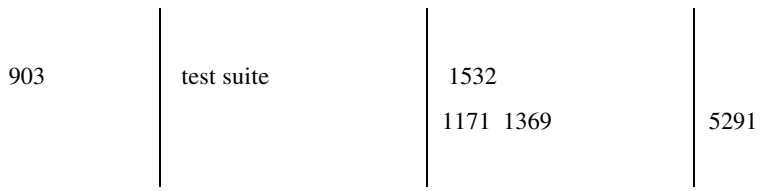

FIGURE AÑ15/X.403

Multiple cross references

A.6.2 *Cross reference listing*

At the end of MTS (P1) and IPMS (P2) test suites there is a separate cross reference listing of all the ASPs, test PDUs and library components together with the place identifiers of all their occurrences in the test suite.

Example:

```
                    :
                    :
                    :
IM–UAPDU–1–14       586 1467
IM–UAPDU–1–15       587 1470
                    :
                    :
                    :
```

The numbers on the right side indicate the places where the item occurs in the test suite.

ANNEX B

(to Recommendation X.403)

IPMS (P2) PICS proformas

B.1 *General*

As a prerequisite to conformance testing, the supplier of an IPMS (P2) implementation must provide a Protocol Implementation Conformance Statement (PICS).

The proforma IPMS (P2) PICS contained in this Annex specifies the information to be supplied.

This information is needed for test case selection. Suppliers should note that tests will be performed to check that services shown as not supported are in fact not present rather than improperly implemented.

The IPMS (P2) PICS is in two parts:

Ñ a part requesting information concerning the support of service elements;

Ñ a part requesting information concerning the support of protocol elements.

Information on service element support is requested in tabular form where, for each service element:

Ñ the status of the service element is indicated as mandatory (M), optional (O) or not applicable (Ñ) in columns labelled ÒSTDÓ;

Ñ the actual support of the service element by the implementation on origination and reception is indicated by the supplier in columns labelled ÒIMPÓ.

Information on protocol element support is requested in tabular form where, for each protocol element:

Ñ the status of the protocol element on origination and reception is indicated as mandatory (M) or optional (O) in columns labelled ÒSTDÓ;

Ñ any implementation constraints are indicated in the column labelled ÒCONST STDÓ where constraints are interpreted as a minimum for reception and a maximum for origination;

Ñ the actual support of the protocol element by the implementation on origination and on reception is indicated by the supplier in the column labelled ÒSTATUS IMPÓ;

Ñ the actual constraints of the implementation on origination and on reception is indicated by the supplier in the columns labelled ÒCONST IMPÓ.

Constraints may be expressed as a length or size (octets, bits, . . .), a value (32k Ñ 1) or a number of occurrences (4) depending on the element being constrained.

B.2 *IPMS (P2) PICS service elements proforma*

The requirements of the X.400 Recommendations are shown in the STD columns of the proforma B.1/X.403 using the following keys:

M Mandatory element (X.401 Basic or Essential Options)

O Optional element (X.401 Additional Options)

Ñ Not applicable service element.

Suppliers of an implementation should use the IMP columns in the proforma to specify information concerning the support of service elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those service elements that are not supported.

B.3 *IPMS(P2) PICS protocol elements proformas*

The requirements of the X.400 Recommendations are shown in the STATUS STD columns of the proformas in Tables BÑ2/X.403 to BÑ4/X.403 using the following keys:

M Mandatory element (X.401 Basic or Essential Optional)

O Optional element (X.401 Additional Optional)

TABLE BÑ1/X.403

IPMS (P2) service elements proforma

SERVICE ELEMENT	ORIGINATION		RECEPTION	
	STD	IMP	STD	IMP
IP Message Identification	M		M	
Typed body	M		M	
Blind copy recipient indication	O		M	
Non receipt notification	O		O	
Receipt notification	O		O	
Auto forwarded indication	O		M	
Originator indication	M		M	
Authorizing users indication	O		M	
PrimaryÑcopy recipients indication	M		M	
Expiry date indication	O		M	
Cross referencing indication	O		M	
Importance indication	O		M	
Obsoleting indication	O		M	
Sensitivity indication	O		M	
Subject indication	M		M	
Replying IP message indication	M		M	
Reply request indication	O		M	
Forwarded IP message indication	O		M	
Body part encryption indication	O		M	
Multipart body	O		M	
Alternate recipient allowed	O		O	
Conversion prohibition	M		M	
Deferred delivery	M		Ñ	
Deferred delivery cancellation	O		Ñ	
Delivery notification	M		Ñ	
Disclosure of other recipients	O		M	
Explicit conversion	O		Ñ	
Grade of delivery selection	M		M	
MultiÑdestination delivery	M		Ñ	
Prevention of nonÑdelivery notification	O		Ñ	
Probe	O		Ñ	
Return of contents	O		Ñ	
Alternate recipient assignment	Ñ		O	
Hold for delivery	Ñ		O	
Implicit conversion	O		O	

Protocol elements which correspond directly to service elements are indicated as mandatory if their corresponding service elements are shown in X.401 (1984) as Basic or Essential Optional, and as optional if their corresponding service elements are shown in X.401 (1984) as Additional Optional. Other protocol elements are indicated as mandatory or optional according to their designation in the UAPDU definitions in X.420 (1984).

The pragmatic constraints of the X.400 Implementor's Guide are shown in the CONST STD columns of the proformas in Tables BÑ2/X.403 to BÑ4/X.403.

Suppliers of an implementation should use:

Ñ the STATUS IMP columns in each proforma to specify information concerning the support of protocol elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those protocol elements that are not supported;

Ñ the CONST IMP columns in each proforma to specify the actual constraints of the implementation.

TABLE BÑ2/X.403

ORDescriptor proforma

ORDescriptor	ORIGINATION			RECEPTION			
	STATUS		CONST	STATUS		CONST	CONST
	STD	IMP	IMP	STD	IMP	IMP	STD Octets
ORName and/or FreeFormName	M			M			
ORName	O			O			
StandardAttributeList	M			M			
CountryName	O			O			
PrintableString	M			M			2
NumericString	M			M			3
ADMDName	O			O			
PrintableString	M			M			16
NumericString	M			M			16
X121Address	O			O			15
TerminalID	O			O			24
PrivateDomainName	O			O			
PrintableString	M			M			16
NumericString	M			M			16
OrganizationName	O			O			64
UniqueUAIdentifier	O			O			32
PersonalName	O			O			64
surname	M			M			40
givenName	O			O			16
initials	O			O			5
generationQualifier	O			O			3
OrganizationalUnit	O			O			32
DomainDefinedAttrList	O			O			4
type	M			M			8
value	M			M			128
FreeFormName	O			O			128
TelephoneNumber	O			O			32

TABLE BÑ3/X.403

IMÑUAPDU proforma

UAPDU NAME: IMÑUAPDU	ORIGINATION			RECEPTION			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
HEADING	M			M			
IPMessageID	M			M			
ORName	O			O			
PrintableString	M			M			64
Originator (ORDescr.)	M			M			
AuthorizingUser (ORDescr.)	O			M			
PrimaryRecipient (ORDescr)	M			M			
ReportRequest	O			O			
ReplyRequest	O			O			
CopyRecipient (ORDescr.)	M			M			
ReportRequest	O			O			
ReplyRequest	O			O			
BlindCopyRecipient (ORDes)	O			M			
ReportRequest	O			O			
ReplyRequest	O			O			
InReplyTo (IPMessageId)	M			M			
ORName	O			O			
PrintableString	M			M			64
Obsoletes (IPMessageId)	O			M			
ORName	O			O			
PrintableString	M			M			64
CrossReference (IPM.Id)	O			M			
ORName	O			O			
PrintableString	M			M			64
Subject	M			M			256
ExpiryDate	O			M			
ReplyBy	O			M			
ReplyToUser (ORDescr.)	O			M			
Importance	O			M			
Sensitivity	O			M			
Autoforwarded	O			M			
Body	M			M			
IA5Text	O			O			
repertoire	O			O			
IA5String	M			M			

TABLE BÑ3/X.403 *(suite)*

UAPDU NAME: IMÑUAPDU	ORIGINATION			RECEPTION			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
G3Fax	O			O			
NumberOfPages	O			O			
G3NonBasicParams	O			O			
BitString	M			M			
TTX	O			O			
NumberOfPages	O			O			
TelexCompatible	O			O			
TeletexNonBasicParams	O			O			
GraficCharacterSet	O			O			
PageFormats	O			O			
MiscTerminalCapability	O			O			
PrivateUse	O			O			
T61String	M			M			
TIF.O	O			O			
T73Document	M			M			
TIF.1	O			O			
T73Document	M			M			
Videotex	O			O			
Videotexstring	M			M			
SFD	O			O			
SFD.Document	M			M			
TLX	O			O			
Voice	O			O			
Bit string	M			M			
Encrypted	O			O			
Bit string	M			M			
NationallyDefined	O			O			
ForwardedIpMessage	O			O			
Delivery	O			O			
DeliveryEnvelope	O			O			
ContentType	M			M			
Originator	M			M			
Original	M			M			
Priority	O			O			
DeliveryFlags	M			M			
OtherRecipients	O			O			
ThisRecipient	M			M			
IntendedRecipient	O			O			
Concreted	O			O			
Submission	M			M			
IMÑUAPDU	M			M			

TABLE BÑ4/X.403

SRÑUAPDU proforma

UAPDU NAME: IMÑUAPDU	ORIGINATION			RECEPTION			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
NonReceiptInformation	M			M			
Reason	M			M			
NonReceiptQualifier	O			O			
Comments	O			O			
Returned	O			O			256
ReceiptInformation	M			M			
ReceiptTime	M			M			
TypeOfReceipt	O			O			
SupplementaryInformation	O			O			64
Reported (IPMessageId)	M			M			
ORName	O			O			
PrintableString	M			M			64
ActualRecipient (ORDescr.)	O			O			
IntendedRecipient (ORDescr)	O			O			
Converted	O			O			

ANNEX C

(to Recommendation X.403)

MTS (P1) PICS proformas

C.1 *General*

As a prerequisite to conformance testing, the supplier of an MTS (P1) implementation must provide a Protocol Implementation Conformance Statement (PICS).

The proforma MTS (P1) PICS contained in this Annex specifies the information to be supplied.

This information is needed for test case selection. Suppliers should note that tests will be performed to check that services shown as not supported are in fact not present rather than improperly implemented.

The MTS (P1) PICS is in two parts:

- Ñ a part requesting information concerning the support of service elements;
- Ñ a part requesting information concerning the support of protocol elements.

Information on service element support is requested in tabular form where, for each service element:

- Ñ the status of the service element is indicated as mandatory (M), optional (O) or not applicable (Ñ) in columns labelled ÒSTDÓ;

Ñ the actual support of the service element by the implementation on origination and reception is indicated by the supplier in columns labelled ÒIMPÓ.

Information on protocol element support is requested in tabular form where, for each protocol element:

Ñ the status of the protocol element on origination and reception is indicated as mandatory (M) or optional (O) in columns labelled ÒSTDÓ;

Ñ any implementation constraints are indicated in the column labelled ÒCONST STDÓ where constraints are interpreted as a minimum for reception and a maximum for origination;

Ñ the actual support of the protocol element by the implementation on origination and on reception is indicated by the supplier in the column labelled ÒSTATUS IMPÓ;

Ñ the actual constraints of the implementation on origination and on reception is indicated by the supplier in the columns labelled ÒCONST IMPÓ.

Constraints may be expressed as a length or size (octets, bits, . . .), a value (32k Ñ 1) or a number of occurrences (4) depending on the element being constrained.

C.2 *Originator/recipient/relay capability*

Suppliers of an implementation should specify Originator/Recipient/Relay capabilities in the IMPLEMENTED column of Table CÑ1/X.403.

TABLE CÑ1/X.403

CAPABILITY	IMPLEMENTED
Originator	
Recipient	
Relay	

C.3 *MTS (P1) PICS service elements proforma*

The requirements of the X.400 Recommendations are shown in the STD columns of the proforma using the following keys:

M Mandatory element (X.401 Basic or Essential Optional)

O Optional element (X.401 Additional Optional)

Ñ Not applicable service element.

Suppliers of an implementation should use the IMP columns in the proforma to specify information concerning the support of service elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those service elements that are not supported.

TABLE CÑ2/X.403

MTS (P1) Service elements proforma

SERVICE ELEMENT	ORIGINATION		RECEPTION		RELAY	
	STD	IMP	STD	IMP	STD	IMP
Content type indication	M		M		Ñ	
Convert indication	M		M		M	
Delivery time stamp indication	M		M		Ñ	
Message identification	M		M		Ñ	
NonÑdelivery notification	M		M		M	
Original encoded information types indication	M		M		Ñ	
Registered encoded information types	M		M		Ñ	
Submission time stamp indication	M		M		Ñ	
Alternate recipient allowed	M		O		Ñ	
Deferred delivery	M		Ñ		Ñ	
Deferred delivery cancellation	M		Ñ		Ñ	
Delivery notification	M		M		Ñ	
Disclosure of other recipients	M		M		M	
Grade of delivery selection	M		M		Ñ	
Multidestination delivery	M		M		M	
Prevention of nonÑdelivery notification	O		O		O	
Return of contents	O		O		O	
Conversion prohibition	M		M		Ñ	
Explicit conversion	O		O		O	
Implicit conversion	O		O		O	
Probe	M		M		M	
Hold for delivery	Ñ		O		Ñ	
Alternate recipient assignment	Ñ		O		Ñ	

C.4 *MTS (P1) protocol elements proformas*

The requirements of the X.400 Recommendations are shown in the STATUS STD column of the proformas in Tables CÑ3/X.403 to CÑ6/X.403 using the following keys:

M Mandatory element (X.401 Basic or Essential Optional)

O Optional element (X.401 Additional Optional)

In the tables below, protocol elements which correspond directly to service elements are indicated as mandatory if their corresponding service elements are shown in X.401 (1984) as Basic or Essential Optional, and as optional if their corresponding service elements are shown in X.401 (1984) as Additional Optional. Other protocol elements are indicated as mandatory or optional according to their designation in the MPDU definitions in X.411 (1984).

For relay functions, protocol elements are indicated as mandatory or optional based only on their status in the P1 protocol specification.

The pragmatic constraints of the X.400 Implementor's Guide are shown in the CONS STD columns of the proformas in Tables CÑ3/X.403 to CÑ6/X.403.

Suppliers of an implementation should use:

Ñ the STATUS IMP column in each proforma to specify information concerning the support of protocol elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those protocol elements that are not supported;

Ñ the CONS IMP columns in each proforma to specify the actual constraints of the implementation.

TABLE CÑ3/X.403

ORName and EncodedInformationType proforma

	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
ORName	M			M			M			
StandardAttributeList	M			M			M			
CountryName	O			O			O			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
X121Address	O			O			O			15 Ch
TerminalID	O			O			O			24 Ch
PrivateDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
OrganizationName	O			O			O			64 Ch
UniqueUAIdentifier	O			O			O			32 Ch
PersonalName	O			O			O			64 Ch
surname	M			M			M			40 Ch
givenName	O			O			O			16 Ch
initials	O			O			O			5 Ch
generationQualifier	O			O			O			3 Ch
OrganizationalUnit	O			O			O			32 Ch
DomainDefinedAttributeList	O			O			O			8 Ch
type	M			M			M			
value	M			M			M			128 Ch
EncodedInformationType										
BITSTRING	M			M			M			32 bit
G3NonBasisParams	O			O			O			
TeletexNonBasisParams	O			O			O			
PresentationCapabilities	O			O			O			

TABLE CÑ4/X.403

UserMPDU proforma

MPDU NAME: UserMPDU	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
UMPDU ENVELOPE	M			M			M			
MPDUIdentifier	M			M			M			
GlobalDomainIdentifier	M			M			M			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
PrivateDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
IA5String	M			M			M			32 Ch
Originator	M			M			M			
OriginalEncodedInformationType	M			M			M			
ContentType	M			M			M			16 Ch
UaContentID	O			O			O			
Priority	M			M			M			
PerMessageFlag	M			M			M			16 b
DisclosureRecipients	M			M			M			
ConversionProhibited	M			M			M			
AlternateRecipientAllowed	M			O			O			
ContentReturnRequest	O			O			O			
DeferredDelivery	M			O			O			
PerDomainBilateralInfo	O			O			O			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
BilateralInfo	M			M			M			a)

TABLE CÑ4/X.403 *(suite)*

MPDU NAME: UserMPDU	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
RecipientInfo	M			M			M			
Recipient	M			M			M			
ExtensionIdentifier	M			M			M			b)
PerRecipientFlag	M			M			M			
ResponsabilityFlag	M			M			M			
ReportRequest	M			M			M			
UserReportRequest	M			M			M			
ExplicitConversion	O			O			O			
TraceInformation	M			M			M			
GlobalDomainIdentifier	M			M			M			
DomainSuppliedInfo	M			M			M			
Arrival	M			M			M			
Deferred	M			M			M			
Action relayed or rerouted	M			O			O			
Converted	O			O			O			
Previous	O			O			O			
UMPDUÑCONTENT	M			M			M			

a) 1024 octets.

b) Max value 32k Ñ 1.

TABLE CÑ5/X.403

DeliveryReportMPDU proforma

MPDU NAME: DeliveryReportMPDU	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
DELIVERYREPORTENVELOPE	M			M			M			
Report	M			M			M			
GlobalDomainIdentifier	M			M			M			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
PrivateDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
IA5String	M			M			M			32 Ch
Originator	M			M			M			
TraceInformation	M			M			M			
GlobalDomainIdentifier	M			M			M			
DomainSuppliedInfo	M			M			M			
Arrival	M			M			M			
Deferred	M			M			M			
Action relayed or rerouted	M			O			O			
Converted	O			O			O			
Previous	O			O			O			
DELIVERYREPORTCONTENT	M			M			M			
Original MPDUIdentifier	M			M			M			
GlobalDomainIdentifier	M			M			M			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
PrivateDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
IA5String	M			M			M			32 Ch
Intermediate	O			O			O			
TraceInformation	M			M			M			
GlobalDomainIdentifier	M			M			M			
DomainSuppliedInfo	M			M			M			
Arrival	M			M			M			
Deferred	M			M			M			
Action relayed or rerouted	M			O			O			
Converted	O			O			O			
Previous	O			O			O			

TABLE CÑ5/X.403 *(suite)*

MPDU NAME: DeliveryReportMPDU	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
UAContentId	O			O			O			
ReportRecipientInfo	M			M			M			
Recipient	M			M			M			
ExtensionIdentifier	M			M			M			32k Ñ1
PerRecipientFlag	M			M			M			
ResponsabilityFlag	M			M			M			
ReportRequest	M			M			M			
UserReportRequest	M			M			M			
LastTraceInformation	M			M			M			
arrival	M			M			M			
converted	O			O			O			
Report	M			M			M			
DeliveredInfo	M			M			M			
Delivery	M			M			M			
TypeOfUA	O			O			O			
NonDeliveredInfo	M			M			M			
ReasonCode	M			M			M			
DiagnosticCode	O			O			O			
IntendedRecipient	O			O			O			
SupplementaryInform.	O			O			O			64 Ch
Returned	O			O			O			
BillingInformation	O			O			O			[a]

[a] 1024 octets.

TABLE CÑ6/X.403

ProbeMPDU proforma

MPDU NAME: UserMPDU	ORIGINATION			RECEPTION			RELAY			CONST
	STATUS		CONST	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD	IMP	IMP	STD
PROBE ENVELOPE	M			M			M			
Probe	M			M			M			
GlobalDomainIdentifier	M			M			M			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
PrivateDomainName	O			O			O			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
IA5String	M			M			M			32 Ch
Originator	M			M			M			
ContentType	M			M			M			
UAContentId	O			O			O			
OriginalEncodedInformationType	M			M			M			
TraceInformation	M			M			M			
GlobalDomainIdentifier	M			M			M			
DomainSuppliedInfo	M			M			M			
Arrival	M			M			M			
Deferred	M			M			M			
Action relayed or rerouted	M			O			O			
Converted	O			O			O			
Previous	O			O			O			
PerMessageFlag	M			M			M			
DisclosureRecipients	M			M			M			
ConversionProhibited	M			M			M			
AlternateRecipientAllowed	M			O			O			
ContentReturnRequest	O			O			O			
ContentLength	O			O			O			
PerDomainBilateralInfo	O			O			O			
CountryName	M			M			M			
NumericString	M			M			M			3 Ch
PrintableString	M			M			M			3 Ch
AdministrationDomainName	M			M			M			
NumericString	M			M			M			16 Ch
PrintableString	M			M			M			16 Ch
BilateralInfo	M			M			M			a)
RecipientInfo	M			M			M			
Recipient	M			M			M			
ExtensionIdentifier	M			M			M			b)
PerRecipientFlag	M			M			M			
ResponsibilityFlag	M			M			M			
ReportRequest	M			M			M			
UserReportRequest	M			M			M			
ExplicitConversion	O			O			O			

a) 1024 octets.

b) Max value 32k Ñ 1.

ANNEX D

(to Recommendation X.403)

RTS PICS proformas

D.1 *General*

As a prerequisite to conformance testing of an RTS implementation, the supplier must provide a Protocol Implementation Conformance Statement (PICS).

The proforma RTS PICS contained in this Annex specifies the information to be supplied.

This information is needed for test case selection. Suppliers should note that tests will be performed to check that services shown as not supported are in fact not present rather than improperly implemented.

The RTS PICS is in three parts:

Ñ Two parts requesting information concerning the support of RTS service primitives.

If primitives have only mandatory parameters, they should be marked as Ònot supportedÓ if any of their parameters are not supported.

Ñ A part requesting information concerning the support of protocol elements.

Information on service element support is requested in tabular form where, for each service element:

Ñ the status of the service element is indicated as mandatory (M), optional (O), conditional (C) or not applicable (Ñ) in columns labelled ÒSTDÓ;

Ñ the actual support of the service element by the implementation as initiator or responder is indicated by the supplier in columns labelled ÒIMPÓ.

Information on protocol element support is requested in tabular form where, for each protocol element:

Ñ the status of the protocol element where the IUT is initiator or responder is indicated as mandatory (M) or optional (O) in columns labelled ÒSTDÓ;

Ñ any implementation constraints are indicated in the column labelled ÒCONST STDÓ where constraints are interpreted as a minimum for reception and a maximum for origination;

Ñ the actual support of the protocol element by the implementation as initiator or responder is indicated by the supplier in the column labelled ÒSTATUS IMPÓ;

Ñ the actual constraints of the implementation as initiator or responder are indicated by the supplier in the columns labelled ÒCONST IMPÓ.

Constraints may be expressed as a length or size (octets, bits, . . .) or a value (32) depending on the element being constrained.

D.2 *RTS PICS service primitives proforma*

The requirements of the X.400 Recommendations are shown in the STD columns of the proforma using the following keys:

M Mandatory element

O Optional element

Suppliers of an implementation should use the IMP columns in the proforma to specify information concerning the support of service elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those service primitives that are not supported.

D.3 *RTS PICS service parameters proforma*

RTS service parameters are mapped to Session and Presentation as below:

- Ñ The parameters of the OPEN.Request and the OPEN.Indication are mapped to the SCONNECT.Request and SCONNECT.Indication and to the corresponding PConnect.
- Ñ Responder/InitiatorÑaddress and InitialÑturn are mapped to the SCONNECT.
- Ñ DialogueÑmode, ApplicationÑprotocol and UserÑdata are mapped to the PConnect.
- Ñ The parameters of the OPEN.Response and OPEN.Confirmation are all mapped to PAccept or PRefuse.

TABLE DÑ1/X.403

RTS service	INITIATOR		RESPONDER	
	STATUS		STATUS	
	STD	IMP	STD	IMP
OPEN	M		M	
CLOSE	M		M	
TURNÑGIVE	O		O	
TURNÑPLEASE	O		O	
TRANSFER	M		M	
EXCEPTION	M		M	

Since all OPEN service parameters are mapped to the PConnect protocol element (apart from ResponseÑaddress and InitialÑturn which are mandatory), there is an apparent duplication of information requested in TablesÊDÑ2/X.403 to DÑ5/X.403 with that requested in Table DÑ6/X.403.

Tables DÑ2/X.403 to DÑ5/X.403 are useful nevertheless because they make sure that all the mandatory parameters are really supported and because they make a static conformance review easier.

The requirements of the X.400 Recommendations are shown in the STD columns of the proforma using the following keys:

- M Mandatory parameters
- O Optional parameters
- C Conditional parameters
- Ñ Not applicable service parameters

Suppliers of an implementation should use the IMP columns in the proforma to specify information concerning the support of service elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those service elements that are not supported.

D.4 *RTS protocol elements*

The requirements of the X.400 Recommendations are shown in the STATUS STD column of the proforma in Table DÑ6/X.403 to DÑ9/X.403 using the following keys:

- M Mandatory element
- O Optional element

The pragmatic constraints of the X.400 Implementor's Guide are shown in the CONST STD columns of the proforma in Table DÑ6/X.403 to DÑ9/X.403.

Suppliers of an implementation should use:

Ñ the STATUS IMP column in the proforma to specify information concerning the support of protocol elements. For convenience, it is suggested that suppliers need only indicate with an ÒXÓ those protocol elements that are not supported;

Ñ the CONST IMP columns in the proforma to specify the actual constraints of the implementation.

TABLE DÑ2/X.403

	INITIATOR	
OPEN.Request	STATUS	
	STD	IMP
OPEN.Request		
ResponderÑaddress	M	
DialogueÑmode	M	
monologue	M	
twa	O	
InitialÑturn	M	
initiator	M	
responder	Ñ	
ApplicationÑprotocol	M	
P1	M	
UserÑdata	C	

TABLE DÑ3/X.403

	INITIATOR	
OPEN.Request	STATUS	
	STD	IMP
OPEN.Request		
ResponderÑaddress	M	
DialogueÑmode	M	
monologue	M	
twa	O	
InitialÑturn	M	
initiator	M	
responder	Ñ	
ApplicationÑprotocol	M	
P1	M	
UserÑdata	C	

TABLE DÑ4/X.403

	INITIATOR	
OPEN.Response	STATUS	
	STD	IMP
OPEN.Response		
Disposition	M	
accepted	C	
refused	C	
UserÑdata	C	
RefusalÑreason	C	
unacceptable dialogue mode	C	
authentication failure	C	
busy	C	

TABLE DÑ5/X.403

	INITIATOR	
OPEN.Confirmation	STATUS	
	STD	IMP
OPEN.Request		
Disposition	M	
accepted	C	
refused	C	
UserÑdata	C	
RefusalÑreason	C	
unacceptable dialogue mode	C	
authentication failure	C	
busy	C	

For some parameters, only one value is applicable (e.g. DataTransferSyntax: O). There are other parameters (e.g. checkpointSize, RefuseReason) that may vary under various circumstances and runÑtime conditions. This information is available in a PIXIT and in such cases a reference to the PIXIT normally can be made in the constraintsÑfield, if the parameter is not fixed.

In a recovery, the SessionConnectedId is used in the PConnect and the PAccept. This SessionConnectedID may or may not be encoded according to X.409. This information is not important for the PICS because it is not a criterion for the Static Conformance Review or for the Test Case Selection and would normally be given in a PIXIT.

TABLE DÑ6/X.403

PConnect	INITIATOR			RESPONDER			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
PConnect	M			M			
DataTransferSyntax	M			M			
pUserData	M			M			
checkpointSize	O			M			
windowSize	O			M			
dialogueMode	O			M			
monologue	M			M			
twa	O			M			
ConnectionData	M			M			512
open	M			M			
null	M			M			
MTAName	M			M			32 oct
Password	M			M			64 oct
recover	O			M			
SessionConnectionIden.	M			M			
CallingSSUserReferen.	M			M			64 oct
CommonReference	M			M			
AdditionalRef.Info.	O			O			4 oct
applicationProtocol	O			M			

TABLE DÑ7/X.403

PAccept	INITIATOR			RESPONDER			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
PAccept	M			M			
DataTransferSyntax	M			M			
PuserData	M			M			
checkpointSize	O			O			
windowSize	O			O			
ConnectionData	M			M			512
open	M			M			
null	M			M			
MTAName	M			M			32 oct
Password	M			M			64 oct
recover	O			O			
SessionConnectionIden.	M			M			
CallingSSUserReferen.	M			M			64 oct
CommonReference	M			M			
AdditionalRef.Info.	O			O			4 oct

TABLE DÑ8/X.403

PRefuse	INITIATOR			RESPONDER			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
PRefuse	M			M			
RefuseReason	M			M			
rtsBusy	C			M			
cannotRecover	C			M			
validationFailure	C			M			
unacceptableDialogueMode	C			M			

TABLE DÑ9/X.403

AbortInformation	INITIATOR			RESPONDER			CONST
	STATUS		CONST	STATUS		CONST	
	STD	IMP	IMP	STD	IMP	IMP	STD
AbortInformation	M			M			
AbortReason	O			O			
localSystemProblem	C			M			
invalidParameter	C			M			
unrecognizedActivity	C			M			
temporaryProblem	C			M			
protocolError	C			M			
transferCompleted	C			M			
reflectedParameter	O			O			