INTERNATIONAL TELECOMMUNICATION UNION

CCITT

X.227

(09/92)

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

DATA COMMUNICATION NETWORKS

CONNECTION-ORIENTED PROTOCOL SPECIFICATION FOR THE ASSOCIATION CONTROL SERVICE ELEMENT

Recommendation X.227

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.227 was prepared by Study Group VII and was approved under the Resolution No. 2 procedure on the 10th of September 1992.

CCITT NOTES

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized private operating agency.

CONTENTS

INTRODUCTION

This Protocol Specification is one of a set of Recommendations and International Standards produced to facilitate the interconnection of information processing systems. It is related to other CCITT Recommendations and International Standards in the set as defined by the Reference Model for Open Systems Interconnection (CCITT Rec. X.200 | ISO 7498). The Reference Model subdivides the areas of standardization for interconnection into a series of layers of specification, each of manageable size.

The goal of Open Systems Interconnection is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of information processing systems:

- from different manufacturers;
- under different managements;
- of different levels of complexity; and
- of different technologies.

This Protocol Specification specifies the connection-oriented mode protocol for the application-service-element for application-association control: the Association Control Service Element (ACSE). The protocol for the ACSE connectionless mode service (A-UNIT-DATA) is specified in CCITT Rec. X.237 | ISO/IEC 10035.

The ACSE connection-oriented mode provides services for establishing and releasing application-associations. The ACSE protocol also includes an optional functional unit for exchanging information to support authentication during association establishment. The ACSE services apply to a wide range of application-process communication requirements.

This Protocol Specification includes an annex that describes the protocol machine of ACSE in terms of a state table. This protocol machine is referred to as the Association Control Protocol Machine (ACPM).

The protocol defined in this Protocol Specification is also govered by the use of the presentation-service (CCITT Rec. X.216 | ISO/IEC 8822) and the session-service (CCITT Rec. X.215 | ISO 8326).

Quality of Services (QOS) is a parameter of the A-ASSOCIATE service. Work is still in progress to provide an integrated treatment of QOS across all of the layers of the OSI Reference Model and to ensure that the inidividual treatments in each layer service satisfy overall QOS objectives in a consistent manner. As a consequence, an addendum may be added to this Protocol Specification at a later time which reflects further QOS developments and integration.

Recommendation X.227

CONNECTION-ORIENTED PROTOCOL SPECIFICATION FOR THE ASSOCIATION CONTROL SERVICE ELEMENT[1)]

1 Scope

The ACSE supports two modes of communication: connection-oriented and connectionless. The ACSE service definition (CCITT Rec. X.217 | ISO/IEC 8649) includes both modes of communication. This Protocol Specification provides the protocol specification for the connection-oriented mode of communication. The protocol specification for the connectionless mode of communication is contained in CCITT Rec. X.237 | ISO/IEC 10035.

This Protocol Specification defines procedures that are applicable to instances of communication between systems which wish to interconnect in an open systems interconnection environment in a connection-oriented mode. The Protocol Specification includes the Kernel functional unit that is used to establish and release application-associations. The Authentication functional unit provides additional facilities for exchanging information in support of authentication during association establishment without adding new services. The ACSE authentication facilities can be used to support a limited class of authentication methods.

This Protocol Specification specifies:

a) procedures for the transfer of information for application-association control and the authentication of application-entities; and

b) the abstract syntax for the representation of the ACSE APDUs.

The ACSE procedures are defined in terms of:

a) the interactions between peer ACSE protocol machines through the use of presentation-services; and

b) the interaction between an ACSE protocol machine and its service-user.

This Protocol Specification also specifies conformance requirements for systems implementing these procedures. It does not contain tests which can be used to demonstrate conformance.

2 Normative references

The following CCITT Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendation and Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the Recommendations and Standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards. The CCITT Secretariat maintains a list of currently valid CCITT Recommendations.

2.1 *Identical Recommendations | International Standards*

– CCITT Recommendation X.665 (1992) | ISO/IEC 9834-6:1993, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI registration authorities: application processes and application entities.*

---

1) Recommendation X.227 and ISO/IEC 8650 (Information technology – Open Systems Interconnection – Protocol Specification for the Association Control Service Element) are technically equivalent.

2.2 *Paired Recommendations | International Standards equivalent in technical content*

– CCITT Recommendation X.200 (1984), *Reference Model of Open Systems Interconnection for CCITT applications*.

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

– CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1)*.

ISO 8824:1987, *Information processing systems – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.

– CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)*.

ISO 8825:1987, *Information processing systems – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

– CCITT Recommendation X.210 (1988), *Open Systems Interconnection – Layer Service Definition Conventions*.

ISO/TR 8509:1987, *Information processing systems – Open Systems Interconnection – Service conventions*.

– CCITT Recommendation X.215 (1988), *Session service definition for Open Systems Interconnection for CCITT applications*.

ISO 8326:1987, *Information processing systems – Open Systems Interconnection – Basic connection oriented session service definition*.

– CCITT Recommendation X.216 (1988), *Presentation service definition for Open Systems Connection for CCITT applications*.

ISO 8822:1988, *Information processing systems – Open Systems Interconnection – Connection oriented presentation service definition*.

– CCITT Recommendation X.217 (1992), *Service definition for the Association Control Service Element.*

ISO/IEC 8649:1993, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element.*

– CCITT Recommendation X.225 (1988), *Session protocol specification for Open Systems Interconnection for CCITT applications*.

ISO 8327:1987, *Information processing systems – Open Systems Interconnection – Basic connection oriented session protocol specification*.

– CCITT Recommendation X.237 (1992), *Connectionless protocol specification for the Association Control Service Element.*

ISO/IEC 10035:1993, *Information processing systems – Open Systems Interconnection – Connectionless protocol specification for the Association Control Service Element.*

– CCITT Recommendation X.650 (1992), *Open Systems Interconnection (OSI) Reference model for naming and addressing.*

ISO 7498-3:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 3: Naming and addressing.*

– CCITT Recommendation X.660 (1992), *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities – General procedures*.

ISO/IEC 9834-1:1991, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities – General procedures*.

– CCITT Recommendation X.800 (1991), *Security architecture for open systems interconnection for CCITT application.*

ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security architecture.*

2.3 *Additional references*

- CCITT Recommendation X.410 (1984), *Message Handling Systems: Remote Operation and Reliable Transfer Server*.
- ISO 6523:1984, *Data interchange – Structures for the identification of organizations*.
- ISO/IEC 9545:1989, *Information technology – Open Systems Interconnection – Application Layer Structure*.
- ISO/IEC 9545/Amd.1. . .[2], *Information technology – Open Systems Interconnection – Amendment to Application Layer Structure covering extensions*.

3 Definitions

3.1 *Reference Model definitions*

3.1.1 *Basic Reference Model definitions*

This Protocol Specification is based on the concepts developed in CCITT Rec. X.200 | ISO 7498 and makes use of the following terms defined in it:

a) Application Layer;
b) application-process;
c) application-entity;
d) application-service-element;
e) application-protocol-data-unit;
f) application-protocol-control-information;
g) presentation-service;
h) presentation-connection;
i) session-service;
j) session-protocol; and
k) session-connection.

3.1.2 *Security architecture definitions*

This Protocol Specification makes use of the following term defined in CCITT Rec. X.800 | ISO 7498-2:

password.

3.1.3 *Naming and addressing definitions*

This Protocol Specification makes use of the following terms defined in CCITT Rec. X.650 | ISO 7498-3:

a) application-process title;
b) application-entity qualifier;
c) application-entity title[3];
d) application-process invocation-identifier;
e) application-entity invocation-identifier; and
f) presentation address.

---

[2] Presently at the stage of draft.

[3] As defined in CCITT Rec. X.650 | ISO 7498-3, an application-entity title is composed of an application-process title and an application-entity qualifier. The ACSE protocol provides for the transfer of an application-entity title value by the transfer of its component values.

3.2 *Service conventions definitions*

This Protocol Specification makes use of the following terms defined in CCITT Rec. X.210 | ISO/TR 8509:

a) service-provider;

b) service-user;

c) confirmed service;

d) non-confirmed service;

e) provider-initiated service;

f) primitive;

g) request (primitive);

h) indication (primitive);

i) response (primitive); and

j) confirm (primitive).

3.3 *Presentation service definitions*

This Protocol Specification makes use of the following terms defined in CCITT Rec. X.216 | ISO 8822:

a) abstract syntax;

b) abstract syntax name;

c) default context;

d) defined context set;

e) functional unit [presentation];

f) normal mode [presentation];

g) presentation context;

h) presentation data value; and

i) X.410-1984 mode [presentation].

3.4 *Application Layer Structure definitions*

This Protocol Specification makes use of the following terms defined in ISO/IEC 9545 and ISO/IEC 9545 Amd 1:

a) application-context;

b) application-entity invocation;

c) control function; and

d) application-service object.

3.5 *ACSE service definitions*

This Protocol Specification makes use of the following terms defined in CCITT Rec. X.217 | ISO/IEC 8649:

a) application-association; association;

b) Association Control Service Element;

c) ACSE service-user;

d) ACSE service-provider;

e) requestor;

f) acceptor;

g) association-initiator;

h) association-responder;

j) authentication;

j) authentication-function;

k) authentication-value;

l) authentication-mechanism;

m) normal mode;

n) X.410-1984 mode; and

o) disrupt.

3.6 *Association Control protocol specification definitions*

The following terms are introduced in this Protocol Specification:

3.6.1 **Association Control Protocol Machine:** The protocol machine for the Association Control Service Element specified in this Protocol Specification.

3.6.2 **requesting Association Control Protocol Machine:** The Association Control Protocol Machine whose service-user is the requestor of a particular Association Control Service Element service.

3.6.3 **accepting Association Control Protocol Machine:** The Association Control Protocol Machine whose service-user is the acceptor for a particular Association Control Service Element service.

4 Abbreviations

4.1 *Data units*

APDU application-protocol-data-unit

4.2 *Types of application-protocol-data-units*

The following abbreviations have been given to the application-protocol-data-units defined in this Protocol Specification:

AARQ A-ASSOCIATE-REQUEST APDU

AARE A-ASSOCIATE-RESPONSE APDU

RLRQ A-RELEASE-REQUEST APDU

RLRE A-RELEASE-RESPONSE APDU

ABRT A-ABORT APDU

4.3 *Other abbreviations*

The following abbreviations are used in this Protocol Specification:

ACPM Association Control Protocol Machine

ACSE Association Control Service Element

AE application-entity

AEI application-entity invocation

AP application-process

APCI application-protocol-control-information

ASE application-service-element

ASO application-service-object

ASN.1 Abstract Syntax Notation One

CF Control function

cnf confirm primitive

ind indication primitive

OSI Open Systems Interconnection

req request primitive

ROA recognized operating agency

QOS quality of service

5 Conventions

This Protocol Specification employs a tabular presentation of its APDU fields. In § 7, tables are presented for each ACSE APDU. Each field is summarized using the following notation:

M presence is mandatory

O presence is ACPM option

U presence is ACSE service-user option

req source is related request primitive

ind sink is related indication primitive

rsp source is related response primitive

cnf sink is related confirm primitive

ACPM source or sink is the ACPM

The structure of each ACSE APDU is specified in § 9 using the abstract syntax notation of ASN.1 (CCITT Rec. X.208 | ISO 8824).

6 Overview of the protocol

6.1 *Service provision*

The protocol in this Protocol Specification provides the connection-oriented services defined in CCITT Rec. X.217 | ISO/IEC 8649. Both the connection-oriented and connectionless services are listed in Table 1/X.227. The protocol for the connectionless A-UNIT-DATA service is specified in CCITT Rec. X.237 | ISO/IEC 10035.

For a particular association, the ACSE connection-oriented services operate either in the normal mode or in the X.410-1984 mode. The mode of operation is determined by the Mode parameter on the A-ASSOCIATE request primitive.

TABLE 1/X.227

ACSE-services

Communication mode	Service	Type
Connection-oriented	A-ASSOCIATE A-RELEASE A-ABORT A-P-ABORT	Confirmed Confirmed Non-confirmed Provider-initiated
Connectionless	A-UNIT-DATA	Non-confirmed

6.2 *Functional units*

Functional units are used by this Protocol Specification to negotiate ACSE user requirements during association establishment. Two functional units are defined:

a) Kernel functional unit; and

b) Authentication functional unit.

The ACSE Requirements fields on the AARQ and AARE APDUs are used to select the Authentication functional units for the association.

The Kernel functional unit is always available. It is the default functional unit. To be included, the Authentication functional unit shall be explicitly requested on the AARQ APDU and accepted on the AARE APDU.

The selection of the Authentication functional unit supports additional fields on the AARQ, AARE, and ABRT APDUs. It does not affect the elements of procedure. Table 2/X.227 shows the services, APDUs and APDU fields associated with the ACSE functional units.

TABLE 2/X.227

Functional unit APDUs and their fields

Functional Unit	Service	APDU	Field Name
Kernel	A-ASSOCIATE	AARQ	Protocol Version Application Context Name Calling AP Title Calling AE Qualifier Calling AP Invocation-identifier Calling AE Invocation-identifier Called AP Title Called AE Qualifier Called AP Invocation-identifier Called AE Invocation-identifier Implementation Information User Information
		AARE	Protocol Version Application Context Name Responding AP Title Responding AE Qualifier Responding AP Invocation-identifier Responding AE Invocation-identifier Result Result Source-Diagnostic Implementation Information User Information
	A-RELEASE	RLRQ	Reason User Information
		RLRE	Reason User Information
	A-ABORT	ABRT	Abort Source User Information
Authentication	A-ASSOCIATE	AARQ	ACSE Requirements Authentication-mechanism Name Authentication-value
		AARE	Ditto
	A-ABORT	ABRT	Diagnostic

6.3 *Use of the presentation-service*

ACSE's use of the presentation-service is determined by ACSE's mode of operation for an association as specified below.

a) *ACSE normal mode:* The ACPM uses the normal mode of the presentation-service (CCITT Rec. X.216 | ISO 8822). The ACPM uses the presentation-service Kernel functional unit to exchange its APCI and, optionally, ACSE service-user information (i.e. ACSE APDUs) with its peer. The use of additional presentation-service functional units is an ACSE service-user choice. This choice does not affect the operation of the ACPM.

b) *ACSE X.410-1984 mode:* The ACPM uses the X.410-1984 mode of the presentation-service. Only the Kernel functional unit is available when using the presentation-service X.410-1984 mode. In this mode, the ACPM does not exchange its own APCI with its peer. It simply passes through information supplied to it by the ACSE service-user or by the presentation-service.

This Protocol Specification assumes that the ACPM is the sole user of the P-CONNECT, P-RELEASE, P-U-ABORT, and P-P-ABORT services. The ACSE neither uses nor constrains the use of any other presentation service.

When supported by version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the presentation-service is subject to length restrictions for its user-data parameters. This Protocol Specification assumes that a local mechanism detects violations of these constraints and makes the ACSE service-user aware of them. An encoding optimization is specified for A-ABORT to mitigate this problem (see § 7.3.3.1).

6.4 *Relationship to the session-service*

The session functional units required for the session-connection that supports the presentation-connection (that in turn supports the association) are determined by the A-ASSOCIATE service requestor and acceptor. They accomplish this using the Session Requirements parameter on the A-ASSOCIATE primitives. The session functional units are described in CCITT Rec. X.215 | ISO 8326.

The rules of the session-service affect the operation of the ACPM and its service-user. The ACSE service-user must be aware of these constraints. This Protocol Specification assumes that a local mechanism enforces them. Some examples of session-service constraints that affect the ACSE service-user are:

a) the availability of negotiated release; and

b) the possibility of release collisions.

6.5 *Model*

The Association control Protocol Machine (ACPM) is modeled as a finite state machine whose specification is given in this Protocol Specification. The ACPM communicates with its service-user by means of the ACSE service primitives defined in CCITT Rec. X.217 | ISO/IEC 8649. The ACPM communicates with its presentation service-provider by means of the presentation services defined in CCITT Rec. X.216 | ISO 8822.

Note – An application ASE specification that references ACSE need not specify the use of ACSE service primitive parameters that are irrelevant to its operation. The control function (CF) that references ACSE can be modeled to pass such parameters between the ACPM and that part of the AEI to which the parameters are relevant.

The ACPM is driven by the receipt of input events from its ACSE service-user and from its presentation service-provider for the underlying presentation-connection that supports the association. The input events from the ACSE service-user are ACSE request and response primitives. The input events from its presentation service-provider are presentation indication and confirm primitives.

The ACPM responds to input events by issuing output events to its presentation service-provider and to its ACSE service-user. The output events to its presentation service-provider are presentation request and response primitives. The output events to its ACSE service-user are ACSE indication and confirm primitives.

The receipt of an input event, the generation of dependent actions, and the resultant output event are considered to be an indivisible action.

During the establishment of an association between two AEs, the existence of invocations of both the requesting and responding AEs is presumed. How they are created is outside of the scope of this Protocol Specification.

A new invocation of an ACPM is employed upon the receipt of an A-ASSOCIATE request primitive or a P-CONNECT indication primitive. Each such invocation controls exactly one association.

Note – Each association may be identified in an end system by a local mechanism so that the ACSE service-user and the ACPM can refer to the association.

The ACPM is modeled to operate in either one of two modes for a given association: the normal mode; and the X.410-1984 mode as specified below.

a) When operating in the normal mode, an ACPM communicates with its peer ACPM in support of an association by transferring ACSE application protocol data units (APDUs) defined in § 9[4)]. An ACSE APDU is transferred as a presentation data value in the User Data parameter of the presentation primitive used on the underlying presentation-connection.

b) When operating in the X.410-1984 mode, an ACPM does not transfer ACSE APDUs with its peer. In this situation, the sending and receiving of presentation primitives are in themselves significant protocol events.

7 Elements of procedure

The ACSE protocol consists of the following procedures:

a) association establishment;

b) normal release of an association; and

c) abnormal release of an association.

In this clause, a summary of each of these elements of procedure is presented. This consists of a summary of the relevant APDUs, and a high-level overview of the relationship between the ACSE services, the APDUs involved, and the presentation service that is used. § 8 describes how the parameters of the presentation primitives are used. In § 9, a detailed specification of the ACSE APDUs is given using the notation of ASN.1 (CCITT Rec. X.208 | ISO 8824). Annex A presents the state table for the ACPM.

7.1 *Association establishment*

7.1.1 *Purpose*

The association establishment procedure is used to establish an association between two AEs. It supports the A-ASSOCIATE service.

7.1.2 *APDUs used*

The association establishment procedure uses the A-ASSOCIATE-REQUEST (AARQ) and the A-ASSOCIATE-RESPONSE (AARE) APDUs. The fields of the AARQ APDU are listed in Table 3/X.227. The fields of the AARE APDU are listed in Table 4/X.227.

---

4) This is true with one exception. If the association is supported by version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the requesting ACPM does not pass ACSE APCI as user data on a P-U-ABORT request primitive. The absence of ACSE APCI in this situation does not imply that the association is operating in the X.410-1984 mode (see § 7.3.3.1).

7.1.3 *Association establishment procedure*

This procedure is driven by the following events:

a) an A-ASSOCIATE request primitive from the requestor;

b) an AARQ APDU as user data on a P-CONNECT indication primitive;

c) an A-ASSOCIATE response primitive from the acceptor; and

d) a P-CONNECT confirm primitive (that may or may not contain an AARE APDU).

7.1.3.1 *A-ASSOCIATE request primitive*

The requesting ACPM forms an AARQ APDU from parameter values of the A-ASSOCIATE request primitive and optionally, the Protocol Version and implementation information. It issues a P-CONNECT request primitive also using information from the A-ASSOCIATE request primitive. The User Data parameter of the P-CONNECT request primitive contains the AARQ APDU.

The requesting ACPM waits for a primitive from the presentation service-provider and does not accept any other primitive from the requestor other than an A-ABORT request primitive.

7.1.3.2 *AARQ APDU*

The accepting ACPM receives an AARQ APDU from its peer as user data on a P-CONNECT indication primitive.

The ACPM determines if the AARQ APDU is acceptable based on the rules for extensibility (see § 7.4). If the AARQ APDU is not acceptable, a protocol error results (see § 7.3.3.4). The association establishment procedure is disrupted. An A-ASSOCIATE indication primitive is not issued. The association is not established.

TABLE 3/X.227

AARQ APDU fields

Field name	Presence	Source	Sink
Protocol Version	O	ACPM	ACPM
Application Context Name	M	req	ind
Calling AP Title	U	req	ind
Calling AE Qualifier	U	req	ind
Calling AP Invocation-identifier	U	req	ind
Calling AE Invocation-identifier	U	req	ind
Called AP Title	U	req	ind
Called AE Qualifier	U	req	ind
Called AP Invocation-identifier	U	req	ind
Called AE Invocation-identifier	U	req	ind
ACSE-requirements	U	req	ind
Authentication-mechanism Name	U	req	ind
Authentication-value	U	req	ind
Implementation Information	O	ACPM	ACPM
User Information	U	req	ind

Note – The Authentication-mechanism Name and Authentication-value fields are only present if the ACSE-requirements field includes the Authentication functional unit.

TABLE 4/X.227

AARE APDU fields

Field name	Presence	Source	Sink
Protocol Version	O	ACPM	ACPM
Application Context Name	M	rsp	cnf
Responding AP title	U	rsp	cnf
Responding AE Qualifier	U	rsp	cnf
Responding AP-Invocation-identifier	U	rsp	cnf
Responding AE-Invocation-identifier	U	rsp	cnf
Result	M	rsp/ACPM	cnf
Result Source – Diagnostic	M	rsp/ACPM	cnf
ACSE-requirements	U	rsp	cnf
Authentication-mechanism Name	U	rsp	cnf
Authentication-value	U	rsp	cnf
Implementation Information	O	ACPM	ACPM
User Information	U	rsp	cnf

Note – The Authentication-mechanism Name and Authentication-value fields are only present if the ACSE-requirements field includes the Authentication functional unit.

The ACPM next inspects the value of the Protocol Version field[5)] of the AARQ APDU. If the ACPM does not support a common protocol version, it forms an AARE APDU with the following assigned fields:

a) Protocol Version field (optional) with the value that indicates the protocol version(s) that it could support (see § 7.1.5.1);

b) Application Context Name field with the same value as on the AARQ APDU;

c) Result field with the value "rejected (permanent)"; and source – Diagnostic field with the values "ACSE service-provider" and "no common ACSE version".

In this case, the ACPM sends the AARE APDU as user data on a P-CONNECT response primitive with a Result parameter that has the value "user rejection". The ACPM does not issue an A-ASSOCIATE indication primitive. The association is not established.

If the P-CONNECT indication primitive and its AARQ APDU are acceptable, the ACPM inspects the ACSE requirements field if it is present. It removes functional units that it does not support. The ACPM then issues an A-ASSOCIATE indication primitive to the acceptor. The A-ASSOCIATE indication primitive parameters are derived from the AARQ APDU and the P-CONNECT indication primitive. The ACPM waits for a primitive from the acceptor.

7.1.3.3 *A-ASSOCIATE response primitive*

When the accepting ACPM receives the A-ASSOCIATE response primitive, the Result parameter specifies whether the service-user has accepted or rejected the association. The ACPM forms an AARE APDU using the A-ASSOCIATE response primitive parameters. The ACPM sets the Result Source – Diagnostic field to "ACSE service-user" and the value derived from the Diagnostic parameter of the response primitive. The AARE APDU is sent as the User Data parameter on the P-CONNECT response primitive.

If the acceptor accepted the association request, the Result parameter on the related P-CONNECT response primitive specifies "acceptance", and the Result field of the outgoing AARE APDU specifies "accepted". The association is established.

If the acceptor rejected the association request, the Result parameter on the related P-CONNECT response primitive specifies "user-rejection", and the Result field of the AARE APDU contains the appropriate rejection value. The association is not established.

5) If the Protocol Version field is not present in the AARQ APDU, version 1 is assumed.

7.1.3.4 *P-CONNECT confirm primitive*

The requesting ACPM receives a P-CONNECT confirm primitive. The following situations are possible:

a) the association has been accepted;

b) the accepting ACPM or the acceptor has rejected the association; or

c) the presentation service-provider has rejected the related presentation connection.

If the association was accepted, the P-CONNECT confirm primitive Result parameter specifies "acceptance". The User Data parameter contains an AARE APDU. The Result field of the AARE APDU specifies "accepted". The requesting ACPM issues an A-ASSOCIATE confirm primitive to the requestor derived from parameters from the P-CONNECT confirm primitive and the AARE APDU. The A-ASSOCIATE confirm primitive Result parameter specifies "accepted". The association is established.

If the association was rejected by either the accepting ACPM or by the acceptor, the related P-CONNECT confirm primitive Result parameter specifies "user-rejection". The User Data parameter contains an AARE APDU.

The requesting ACPM issues an A-ASSOCIATE confirm primitive to the requestor derived from parameters from the P-CONNECT confirm primitive and the AARE APDU. The A-ASSOCIATE confirm primitive Result parameter indicates "rejected (transient)" or "rejected (permanent)". The Result Source parameter indicates "ACSE service-user" or "ACSE service-provider". The association is not established.

If the presentation-connection was rejected by the presentation service-provider, the P-CONNECT confirm primitive Result parameter specifies "provider-rejection." In this situation, the User Data field is not used. The requesting ACPM issues an A-ASSOCIATE confirm primitive with the Result parameter indicating "rejected (permanent)." The Result Source parameter indicates "presentation service-provider"[6]. The association is not established.

7.1.4 *Use of the AARQ APDU fields*

The AARQ APDU fields are used by the requesting and accepting ACPMs as specified below.

7.1.4.1 *Protocol Version*

For the requesting ACPM: The value assigned to this field is determined within the implementation of the ACPM. It is a variable length bit string where each bit that is set to one indicates the version of ACSE protocol that this ACPM supports. Bit 0 represents version 1; bit 1 represents version 2; etc. Multiple bits may be set indicating support of multiple versions. No trailing bits higher than the highest version of this Protocol Specification that the requesting ACPM supports are included. That is, the last bit of the string is set to one.

For the accepting ACPM: The ACPM ignores trailing bits of this field that are higher than the one indicating the latest version of this Protocol Specification that it supports.

7.1.4.2 *Application Context Name*

For the requesting ACPM: This value is determined by the value of the Application Context Name parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Application Context Name parameter of the A-ASSOCIATE indication primitive, if issued.

---

6) The presentation-service (CCITT Rec. X.216 | ISO 8822) currently does not define a Diagnostic parameter on the P-CONNECT response. However, work is still in progress to provide an integrated treatment of the "result" related parameters across all layers of the OSI Reference Model. As a consequence, an addendum may be added to this Protocol Specification at a later time that reflects further developments and integration.

7.1.4.3 *Calling AP Title*

For the requesting ACPM: This value is determined by the value of the Calling AP Title parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Calling AP Title parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.4 *Calling AE Qualifier*

For the requesting ACPM: This value is determined by the value of the Calling AE Qualifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Calling AE Qualifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.5 *Calling AP Invocation-identifier*

For the requesting ACPM: This value is determined by the value of the Calling AP Invocation-identifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to derive the value of the Calling AP Invocation-identifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.6 *Calling AE Invocation-identifier*

For the requesting ACPM: This value is determined by the value of the Calling AE Invocation-identifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to derive the value of the Calling AE Invocation-identifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.7 *Called AP Title*

For the requesting ACPM: This value is determined by the value of the Called AP Title parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Called AP Title parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.8 *Called AE Qualifier*

For the requesting ACPM: This value is determined by the value of the Called AE Qualifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Called AE Qualifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.9 *Called AP Invocation-identifier*

For the requesting ACPM: This value is determined by the value of the Called AP Invocation-identifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Called AP Invocation-identifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.10 *Called AE Invocation-identifier*

For the requesting ACPM: This value is determined by the value of the Called AE Invocation-identifier parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Called AE Invocation-identifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.11 *ACSE-requirements*

For the requesting ACPM: The value assigned to this field is determined by the value of the ACSE Requirements parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the ACSE Requirements parameter of the A-ASSOCIATE indication primitive, if issued. The ACPM inspects the ACSE-requirements field and removes any functional units not supported by the ACPM before issuing it to the service-user.

7.1.4.12 *Authentication-mechanism Name*

For the requesting ACPM: The value assigned to this field is determined by the value of the Authentication-mechanism Name parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Authentication-mechanism Name parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.13 *Authentication-value*

For the requesting ACPM: The value assigned to this field is determined by the value of the Authentication-value parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the Authentication-value parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.4.14 *Implementation Information*

For the requesting ACPM: The value assigned to this field is determined within the implementation of the ACPM. It contains information specific to the individual implementation of that ACPM. It is not used in negotiation.

For the accepting ACPM: This field does not affect the operation of the ACPM. Any use depends on a common understanding between the requesting and accepting ACPMs.

7.1.4.15 *User Information*

For the requesting ACPM: This value is determined by the value of the User Information parameter of the A-ASSOCIATE request primitive.

For the accepting ACPM: This value is used to determine the value of the User Information parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.5 *Use of the AARE APDU fields*

The AARE APDU fields are used by the accepting and requesting ACPMs as specified below.

7.1.5.1 *Protocol Version*

For the accepting ACPM: The value of this field assigned by the ACPM depends on whether the association request is accepted or rejected by the ACPM and the acceptor, as specified below.

a) If the association is accepted, the value assigned by the ACPM is a variable length bit string that indicates the protocol version selected by the ACPM from those proposed in the AARQ APDU. Only the bit indicating the version selected is set to one. That bit is the last bit in the string.

b) If the association is rejected, the value assigned by the ACPM is a variable length bit string that indicates the protocol version(s) of this Protocol Specification that could be supported by the ACPM.

For the requesting ACPM: The use of the value in this field depends on whether the association request is accepted or rejected.

a) If the association is accepted, this value defines the protocol version of this Protocol Specification to be used for this association.

b) If the association is rejected, the use of this value is a local option.

7.1.5.2 *Application Context Name*

For the accepting ACPM: This value is determined by the value of the Application Context Name parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Application Context Name parameter of the A-ASSOCIATE confirm primitive.

7.1.5.3 *Responding AP Title*

For the accepting ACPM: This value is determined by the value of the Responding AP Title parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Responding AP Title parameter of the A-ASSOCIATE confirm primitive, if issued.

7.1.5.4 *Responding AE Qualifier*

For the accepting ACPM: This value is determined by the value of the Responding AE Qualifier parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Responding AE Qualifier parameter of the A-ASSOCIATE confirm primitive, if issued.

7.1.5.5 *Responding AP Invocation-identifier*

For the accepting ACPM: This value is determined by the value of the Responding AP Invocation-identifier parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Responding AP Invocation-identifier parameter of the A-ASSOCIATE confirm primitive, if issued.

7.1.5.6 *Responding AE Invocation-identifier*

For the accepting ACPM: This value is determined by the value of the Responding AE Invocation-identifier parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Responding AE Invocation-identifier parameter of the A-ASSOCIATE confirm primitive, if issued.

7.1.5.7 *Result*

For the accepting ACPM: The value is determined by the ACPM or by the acceptor as specified below.

a) If the AARQ APDU is rejected by the ACPM (i.e., an A-ASSOCIATE indication primitive is not issued to the acceptor), the value of "rejected(permanent)" or "rejected (transient)" is assigned by the ACPM.

b) Otherwise, the value is determined by the Result parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Result parameter of the A ASSOCIATE confirm primitive.

7.1.5.8 *Result Source – Diagnostic*

This field contains both the Result Source value and the Diagnostic value.

7.1.5.8.1 *Result Source value*

For the accepting ACPM: This value is assigned by the ACPM as specified below.

a) If the AARQ APDU is rejected by the ACPM (i.e. an A-ASSOCIATE indication primitive is not issued to the acceptor), it assigns the value "ACSE service-provider."

b) Otherwise, the ACPM assigns the value "ACSE service-user."

For the requesting ACPM: This value is used to determine the value of the Result Source parameter of the A-ASSOCIATE confirm primitive.

7.1.5.8.2 *Diagnostic value*

For the accepting ACPM: This value is determined by the ACPM or by the acceptor as specified below.

a) If the AARQ APDU is rejected by the ACPM (i.e. an A-ASSOCIATE indication primitive is not issued to the acceptor), the appropriate value is assigned by the ACPM.

b) Otherwise, the value is determined by the value of the Diagnostic parameter of the A-ASSOCIATE response primitive. If the Diagnostic parameter is not included on the response primitive, the ACPM assigns the value of "null."

For the requesting ACPM: This value is used to determine the value of the Diagnostic parameter of the A-ASSOCIATE confirm primitive, unless it has the value of "null.". In this case, a Diagnostic value is not included.

7.1.5.9 *ACSE-requirements*

For the accepting ACPM: The value assigned to this field is determined by the value of the ACSE Requirements parameter of the A-ASSOCIATE response primitive. This value shall only include functional units that were on the indication primitive.

For the requesting ACPM: This value is used to determine the value of the ACSE Requirements parameter of the A-ASSOCIATE confirm primitive.

7.1.5.10 *Authentication-mechanism Name*

For the accepting ACPM: The value assigned to this field is determined by the value of the Authentication-mechanism Name parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Authentication-mechanism Name parameter of the A-ASSOCIATE confirm primitive.

7.1.5.11 *Authentication-value*

For the accepting ACPM: The value assigned to this field is determined by the value of the Authentication-value parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the Authentication-value parameter of the A-ASSOCIATE confirm primitive.

7.1.5.12 *Implementation Information*

For the accepting ACPM: The value assigned to this field is determined within the implementation of the ACPM. It contains information specific to the individual implementation of that ACPM. It is not used in negotiation.

For the requesting ACPM: This field does not affect the operation of the ACPM. Any use depends on a common understanding between the accepting and requesting ACPMs.

7.1.5.13 *User Information*

For the accepting ACPM: This value is determined by the value of the User Information parameter of the A-ASSOCIATE response primitive.

For the requesting ACPM: This value is used to determine the value of the User Information parameter of the A-ASSOCIATE confirm primitive.

7.1.6 *Collisions and interactions*

7.1.6.1 *A-ASSOCIATE service*

For a given ACPM, an A-ASSOCIATE collision cannot occur (see § 6.5). For a given AE, two distinct ACPMs would be involved that represent the processing for two distinct associations:

a) an ACPM that processes the initial A-ASSOCIATE request primitive that results in the sending of an AARQ as user data on a P-CONNECT request primitive; and

b) an ACPM that processes the subsequently received AARQ APDU as user data on a P-CONNECT indication primitive.

7.1.6.2 *A-ABORT, P-U-ABORT, or P-P-ABORT service*

If an ACPM receives an A-ABORT request primitive, a P-U-ABORT indication primitive, or a P-P-ABORT indication primitive, it discontinues the normal association establishment procedure, and instead follows the abnormal release procedure.

7.2 *Normal release of an association*

7.2.1 *Purpose*

This procedure is used for the normal release of an association by an AE without loss of information in transit. It supports the A-RELEASE service.

7.2.2 APDUs used

The normal release procedure uses the A-RELEASE-REQUEST (RLRQ) APDU and the A-RELEASE-RESPONSE (RLRE) APDU. The fields of the RLRQ APDU are listed in Table 5/X.227. The fields of the RLRE APDU are listed in Table 6/X.227.

TABLE 5/X.227

RLRQ APDU fields

Field name	Presence	Source	Sink
Reason	U	req	ind
User Information	U	req	ind

TABLE 6/X.227

RLRE APDU fields

Field name	Presence	Source	Sink
Reason	U	rsp	cnf
User Information	U	rsp	cnf

7.2.3 *Normal release procedure*

This procedure is driven by the following events:

a) an A-RELEASE request primitive from the requestor;

b) an RLRQ APDU as user data on a P-RELEASE indication primitive;

c) an A-RELEASE response primitive from the acceptor; or

d) an RLRE APDU as user data on a P-RELEASE confirm primitive.

7.2.3.1 *A-RELEASE request primitive*

When an A-RELEASE request primitive is received, the ACPM sends an RLRQ APDU as user data on a P-RELEASE request primitive using the parameters from the A-RELEASE request primitive.

Note – The requestor is required to meet the presentation (and session) requirements in order to issue an A-RELEASE request primitive (see §§ 6.3 and 6.4).

The requesting ACPM now waits for a primitive from the presentation service-provider. It does not accept any primitives from the requestor other than an A-ABORT request primitive.

7.2.3.2 *RLRQ APDU*

When the accepting ACPM receives the RLRQ APDU as user data on a P-RELEASE indication primitive, it issues an A-RELEASE indication primitive to the acceptor. It does not accept any ACSE primitives from its service-user other than an A-RELEASE response primitive or an A-ABORT request primitive.

7.2.3.3 *A-RELEASE response primitive*

The Result parameter on the A-RELEASE response primitive specifies whether the acceptor accepts or rejects the release of the association. The accepting ACPM forms an RLRE APDU from the response primitive parameters. The RLRE APDU is sent as user data on a P-RELEASE response primitive.

a) If the acceptor accepted the release, the Result parameter of the P-RELEASE response primitive has a Result parameter value of “affirmative.” The association is released.

b) If the acceptor rejected the release, the Result parameter of the P-RELEASE response primitive has a Result parameter value of “negative.” The association continues.

Note – To give a negative response, the acceptor is required to meet the related presentation (and session) requirements (see § 6.4).

7.2.3.4 *RLRE APDU*

The requesting ACPM receives a P-RELEASE confirm primitive containing an RLRE APDU from its peer. The Result parameter on the P-RELEASE confirm primitive specifies either that the acceptor agrees or disagrees that the association may be released. The requesting ACPM forms an A-RELEASE confirm primitive from the RLRE APDU fields.

a) If the Result parameter on the P-RELEASE confirm primitive specifies “affirmative”, the association is released.

b) If the Result parameter on the P-RELEASE confirm primitive specifies “negative”, the association continues. The requesting ACPM again accepts primitives from its service-user.

7.2.3.5 *A-RELEASE service collision*

An A-RELEASE service collision occurs when an ACPM has sent out an RLRQ APDU as the user data of a P-RELEASE request primitive (as a result of receiving an A-RELEASE request primitive from its service-user). Instead of receiving the expected RLRE APDU as user data on a P-RELEASE confirm primitive from its peer, it receives an RLRQ APDU as the user data of a P-RELEASE indication primitive.

The ACPM issues an A-RELEASE indication primitive to its service-user. The procedure then followed by an ACPM depends on whether its service-user was the association-initiator or the association-responder.

a) For the association-initiator:

1) The ACPM waits for an A-RELEASE response primitive from its service-user. When it receives the response primitive, it forms an RLRE APDU from the response primitive's parameters. The RLRE is sent as user data on a P-RELEASE response primitive. The association continues.

2) This ACPM now waits for an RLRE from its peer as user data on a P-RELEASE confirm primitive. It does not accept any primitive from its service-user other than an A-ABORT request primitive.

3) When the ACPM receives the RLRE, it forms an A-RELEASE confirm primitive from the RLRE fields and issues it to its service-user. The association is released.

In summary, the sequence of events that drive the ACPM of the association-initiator are:

– A-RELEASE request primitive;

– RLRQ APDU (causing the collision);

– A-RELEASE response primitive; and finally,

– RLRE APDU.

b) For the association-responder:

1) The ACPM waits for an RLRE from its peer as user data on a P-RELEASE confirm primitive. It does not accept a primitive from its service-user other than an A-ABORT request primitive.

2) When this ACPM receives the RLRE, it forms an A-RELEASE confirm primitive from the RLRE fields. The association continues.

3) The ACPM now waits for an A-RELEASE response primitive from its service-user. When it receives the response primitive, it forms a RLRE APDU from the response primitive's parameters. The RLRE is sent as user data on a P-RELEASE response primitive. The association is released.

In summary, the sequence of events that drive the ACPM of the association-responder are:

– A-RELEASE request primitive;

– RLRQ APDU (causing the collision);

– RLRE APDU; and finally

– A-RELEASE response primitive.

7.2.4 *Use of the RLRQ APDU fields*

The RLRQ APDU fields are used by the requesting and accepting ACPMs as specified below.

7.2.4.1 *Reason*

For the requesting ACPM: This value is determined by the value of the Reason parameter of the A-RELEASE request primitive.

For the accepting ACPM: This value is used to determine the value of the Reason parameter of the A-RELEASE indication primitive.

7.2.4.2 *User Information*

For the requesting ACPM: This value is determined by the value of the User Information parameter of the A-RELEASE request primitive.

For the accepting ACPM: This value is used to determine the value of the User Information parameter of the A-RELEASE indication primitive.

7.2.5 *Use of the RLRE APDU fields*

The RLRE APDU fields are used by the accepting and requesting ACPMs as specified below.

7.2.5.1 *Reason*

For the accepting ACPM: This value is determined by the value of the Reason parameter of the A-RELEASE response primitive.

For the requesting ACPM: This value is used to determine the value of the Reason parameter of the A-RELEASE confirm primitive.

7.2.5.2 *User Information*

For the accepting ACPM: This value is determined by the value of the User Information parameter of the A-RELEASE response primitive.

For the requesting ACPM: This value is used to determine the value of the User Information parameter of the A-RELEASE confirm primitive.

7.2.6 *Collisions and interactions*

7.2.6.1 *A-RELEASE service*

For a given ACPM, an A-RELEASE service collision can occur. The processing for such a collision is described in § 7.2.3.5.

Note – An A-RELEASE service collision can only occur if no session tokens were selected for the association.

7.2.6.2 *A-ABORT service, P-U-ABORT, or P-P-ABORT service*

If an ACPM receives an A-ABORT request primitive, a P-U-ABORT indication primitive, or a P-P-ABORT indication primitive, it disrupts the normal association release procedure, and instead follows the abnormal release procedure.

7.3 *Abnormal release of an association*

7.3.1 *Purpose*

The Abnormal Release procedure can be used at any time to force the abrupt release of the association by a requestor in either AE, by either ACPM or by the presentation service-provider. When the abnormal release procedure is applied during an attempt to establish an association, the association is not established. The abnormal release procedure supports the A-ABORT and A-P-ABORT services.

7.3.2 *APDUs used*

The abnormal release procedure uses the A-ABORT (ABRT) APDU. The fields of the ABRT APDU are listed in Table 7/X.227.

Note – No APDUs are defined for the A-P-ABORT service since it is directly mapped from the P-P-ABORT service.

TABLE 7/X.227

ABRT APDU fields

Field name	Presence	Source	Sink
Abort Source	M	ACPM	ind
"Diagnostic	U	req	ind"[1]
User Information	U	req	ind

7.3.3 *Abnormal release procedure*

This procedure is driven by the following events:

a) an A-ABORT request primitive from the requestor;

b) a P-U-ABORT indication primitive;

c) a P-P-ABORT indication primitive; or

d) a protocol error detected by an ACPM.

7.3.3.1 *A-ABORT request primitive*

When an ACPM receives an A-ABORT request primitive from its service-user, the processing that it performs depends on the version of the underlying session-protocol (CCITT Rec. X.225 | ISO 8327) that supports the association as specified below.

a) For version 1, the ACPM does not send any of its APCI to its peer. It simply issues a P-U-ABORT request primitive. If user information is included on the A-ABORT request primitive, that user information is passed as user data on the P-U-ABORT request primitive. The association is released.

b) For other versions, the ACPM sends an ABRT APDU as user data on a P-U-ABORT request primitive. The Abort Source field is specified as "ACSE service-user." If the User Information parameter is included on the A-ABORT request primitive, it is included in the ABRT APDU. The association is released.

7.3.3.2 *P-U-ABORT indication primitive*

When an ACPM receives a P-U-ABORT indication primitive, the User Data parameter may contain[7)] an ABRT APDU.

a) If the indication primitive does not contain an ABRT APDU, the ACPM issues an A-ABORT indication primitive with the Abort Source parameter specified as "ACSE service-user." If user data is contained on the P-U-ABORT indication primitive, it is included as the User Information parameter of the A-ABORT indication primitive. The association is released.

b) If the indication primitive does contain an ABRT APDU, the ACPM issues an A-ABORT indication primitive using the Abort Source field of the ABRT APDU. If a User Information field is contained in the ABRT APDU, it is included on the A-ABORT indication primitive. The association is released.

7.3.3.3 *P-P-ABORT indication primitive*

When an ACPM receives a P-P-ABORT indication primitive, the ACPM issues an A-P-ABORT indication primitive to the acceptor. The association is released.

7.3.3.4 *Protocol errors*

Two types of ACSE protocol errors are possible:

a) for a particular ACPM state, an unexpected APDU is received; or

b) an invalid field is encountered during the processing of an incoming APDU (see § 7.4).

If an unexpected APDU is received, the abnormal release procedure is invoked. If an invalid field is detected by an ACSE procedure, that procedure is disrupted and the abnormal release procedure is invoked.

---

7) If an association is supported by version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the User Data parameter does not contain an ABRT APDU (see § 7.3.3.1). The absence of an APDU in this situation does not imply that the application is operating in the X.410-1984 mode.

As part of the abnormal release procedure, the ACPM issues an A-ABORT indication primitive to its service-user, unless the error occurred during the association establishment procedure[8] as the result of receiving an invalid AARQ (see § 7.4). If an indication primitive is issued, the value of the Abort Source is "ACSE service-provider". The User Information parameter is not used as specified below.

The subsequent ACPM processing performed depends on the version of the underlying session-protocol (CCITT Rec. X.225 | ISO 8327) that supports the association as specified below.

a) For version 1, the ACPM issues a P-U-ABORT request primitive. No user information is included.

b) For other versions, the ACPM sends an ABRT APDU as user data on a P-U-ABORT request primitive. The Abort Source field is specified as "ACSE service-provider". The User Information field is not used.

In either case, the association is released.

7.3.4 *Use of the ABRT APDU fields*

The ABRT APDU fields are used by the requesting and accepting ACPMs as specified below.

7.3.4.1 *Abort Source*

For the requesting ACPM: This value is assigned by the ACPM as specified below.

a) If the ACPM initiated the abort procedure, the ACPM assigns the value of "ACSE service-provider";

b) Otherwise, the ACPM assigns the value of "ACSE service-user".

For the accepting ACPM: This value is used to determine the value of the Abort Source parameter of the A-ABORT indication primitive.

7.3.4.2 *Diagnostic*

For the requesting ACPM: This value is determined by the value of the Diagnostic parameter of the A-ABORT request primitive.

For the accepting ACPM: This value is used to determine the value of the Diagnostic parameter of the A-ABORT indication primitive.

7.3.4.3 *User Information*

For the requesting ACPM: This value is determined by the value of the User Information parameter of the A-ABORT request primitive.

For the accepting ACPM: This value is used to determine the value of the User Information parameter of the A-ABORT indication primitive.

7.3.5 *Collisions and interactions*

The abnormal release procedure may be used whenever an association is established, is in the process of being established, or is being normally released. This procedure disrupts any other currently active procedure. A P-P-ABORT indication primitive can disrupt the A-ABORT procedure with loss of the A-ABORT information. Collisions of ABRT APDUs are governed by the P-U-ABORT services (CCITT Rec. X.216 | ISO 8822).

---

[8] Since an A-ASSOCIATE indication primitive will not be issued, an A-ABORT indication primitive would have no meaning, and, therefore, it is not issued.

7.4 *Rules for extensibility*

When processing an incoming AARQ, the accepting ACPM shall:

a) ignore all tagged values that are not defined in the abstract syntax of this Protocol Specification; and

b) ignore all unknown bit name assignments within a bit string.

After the association has been established or during the establishment of an association, only those ACSE APDUs and related APDU fields defined in the ASN.1 description of the negotiated version of this Protocol Specification shall be issued.

A received APDU or field within an APDU which is not defined in the ASN.1 description of the negotiated version of this Protocol Specification shall be treated as a protocol error.

8 Mapping to the presentation-service

This clause specifies how the presentation service primitives are used by the ACPM. This usage depends on the mode selected (see § 6.3) for the association.

a) *For the requesting ACPM:* The mode for the association is determined by the value of the Mode parameter of the invoking A-ASSOCIATE request primitive. If the Mode parameter is not included on the request primitive, the default value of "normal" is used.

b) *For the accepting ACPM:* The mode is determined by the value of the Mode parameter of the incoming P-CONNECT indication primitive.

Paragraphs 8.1 to 8.3 specify the usage of the presentation services for the normal mode. § 8.4 to § 8.6 specify the usage for the X.410-1984 mode. Table 8/X.227 summarizes the mapping of ACSE primitives and their related APDUs to the presentation primitives used.

TABLE 8/X.227

Mapping overview

ACSE primitive	APDU[a)]	Presentation Primitive
A-ASSOCIATE request/indication A-ASSOCIATE response/confirm	AARQ AARE	P-CONNECT request/indication P-CONNECT response/confirm
A-RELEASE request/indication A-RELEASE response/confirm	RLRQ RLRE	P-RELEASE request/indication P-RELEASE response/confirm
A-ABORT request/indication A-P-ABORT indication	ABRT –	P-U-ABORT request/indication P-P-ABORT indication

a) ACSE APDUs are not used in the X.410-1984 mode.

8.1 *Association establishment (normal mode)*

The association establishment procedure uses the P-CONNECT service. Association establishment takes place concurrently with the establishment of the underlying presentation-connection.

8.1.1 *Directly mapped parameters*

For the P-CONNECT primitives: The following parameters are not referenced by the ACPM and are mapped directly onto the corresponding parameters of the A-ASSOCIATE primitives:

a) Calling Presentation Address;

b) Called Presentation Address;

c) Responding Presentation Address;

d) Presentation Context Definition List;

e) Presentation Context Definition Result List;

f) Default [Presentation] Context Name;

g) Default [Presentation] Context Result;

h) Quality of Service;

i) Presentation Requirements;

j) Session Requirements;

k) Initial Synchronization Point Serial Number;

l) Initial Assignment of Tokens; and

m) Session-connection Identifier.

8.1.2 *Use of other P-CONNECT request and indication parameters*

The Mode and User Data parameters of the P-CONNECT request and indication primitives are referenced by the ACPM.

8.1.2.1 *Mode*

For the P-CONNECT request primitive: The Mode parameter is set to the value of the Mode parameter of the A-ASSOCIATE request primitive. For the normal mode of ACSE operation, this parameter has the value of "normal". This indicates to the presentation-service that it is to operate in the normal mode for this presentation-connection.

For the P-CONNECT indication primitive: This parameter has the value of "normal" for the normal mode of ACSE operation. The value indicates that the accepting ACPM is to operate in the normal mode for this association. The Mode parameter of the A-ASSOCIATE indication primitive is set to the value of "normal".

8.1.2.2 *User Data*

For both the P-CONNECT request and indication primitives: The User Data parameter is used to carry the AARQ APDU as specified below.

a) The APCI of the AARQ APDU is expressed using the ACSE abstract syntax of this Protocol Specification. This abstract syntax must be included as the value of a presentation context definition parameter specified by the requestor on the A-ASSOCIATE request primitive.

   Note – The requesting and accepting ACPMs are aware of the presentation context that contains their abstract syntax by a local mechanism.

b) User information (if any) from the A-ASSOCIATE request primitive is included in the AARQ APDU and is expressed using one or more presentation contexts specified by the requestor on the A-ASSOCIATE request primitive.

8.1.3 *Use of other P-CONNECT response and confirm parameters*

The User Data and Result parameters of the P-CONNECT response and confirm primitive are referenced by the ACPM.

8.1.3.1 *Result*[9)]

For the P-CONNECT response primitive: The Result parameter is set by the accepting ACPM as specified below.

a) If the accepting ACPM itself rejects the association, it is set as "user-rejection".

b) If the accepting ACPM accepts the request, the value is set as "acceptance", or "user-rejection" as determined by the value of the corresponding Result parameter on the A-ASSOCIATE response primitive.

For the P-CONNECT confirm primitive: The Result parameter is used by the requesting ACPM to determine if the P-CONNECT confirm primitive User Data parameter contains an AARE APDU as specified below.

a) If the Result parameter has the value "provider-rejection", the request is rejected by the presentation service-provider. The intended accepting ACPM never received the AARQ APDU. The User Data parameter does not contain an AARE APDU.

b) Otherwise, the Result parameter has the value of "acceptance" or "user rejection." The accepting ACPM received the AARQ APDU and has returned an AARE APDU that is contained in the User Data parameter.

8.1.3.2 *User Data*

The User Data field only has relevance if the P-CONNECT request primitive was not rejected by the presentation service-provider (see § 8.1.3.1).

For both the P-CONNECT response and confirm primitives: The User Data parameter is used to carry the AARE APDU as specified below.

a) The APCI of the AARE APDU is expressed using the ACSE abstract syntax of this Protocol Specification. This abstract syntax must be included as the value of a presentation context definition parameter selected by the acceptor on the A-ASSOCIATE response primitive.

b) User information (if any) from the A-ASSOCIATE response primitive is included in the AARE APDU and is expressed using one or more presentation contexts selected by the acceptor on the A-ASSOCIATE response primitive.

8.2 *Normal release of an association (normal mode)*

The normal release procedure uses the P-RELEASE service. The normal release of an association takes place simultaneously with the normal release of the underlying presentation-connection.

8.2.1 *Use of P-RELEASE request and indication parameters*

The User Data parameter of the P-RELEASE request and indication primitives is referenced by the ACPM.

For both the P-RELEASE request and indication primitives: The User Data parameter is used to carry the RLRQ APDU as specified below.

a) The APCI of the RLRQ APDU is expressed using the ACSE abstract syntax of this Protocol Specification. This abstract syntax must be one of the available presentation contexts.

b) User information (if any) from the A-RELEASE request primitive is included in the RLRQ APDU and is expressed using one or more available presentation contexts.

---

9) The AARE APDU also has a Result field that must correspond to the value of this presentation parameter. The Result parameter of the A-ASSOCIATE confirm primitive is determined by the Result field of the AARE APDU.

8.2.2 *Use of P-RELEASE response and confirm parameters*

The Result and User Data parameters of the P-RELEASE response and confirm primitives are referenced by the ACPM.

8.2.2.1 *Result*

For the P-RELEASE response primitive: The Result parameter is set to the value of the Result parameter of the A-RELEASE response primitive (i.e. "affirmative" or "negative"). This value indicates to the presentation service-provider whether the underlying presentation-connection is to be released or if it is to be continued.

For the P-RELEASE confirm primitive: The value of the Result parameter on the A-ASSOCIATE confirm primitive is set to the value of the Result parameter. This value indicates to the requesting ACPM whether the association is released or if it continues.

8.2.2.2 *User Data*

For both the P-RELEASE response and confirm primitives: The User Data parameter is used to carry the RLRE APDU as specified below.

a) The APCI of the RLRE APDU is expressed using the ACSE abstract syntax of this Protocol Specification. This abstract syntax must be one of the available presentation contexts.

b) User information (if any) from the A-RELEASE response primitive is included in the RLRE APDU and is expressed using one or more available presentation contexts.

8.3 *Abnormal release of an association (normal mode)*

The abnormal release procedure uses the P-U-ABORT and P-P-ABORT services. The abnormal release of an association takes place simultaneously with the abnormal release of the underlying presentation-connection.

8.3.1 *Use of P-U-ABORT request and indication parameters*

The User Data parameter of the P-U-ABORT request and indication primitives is referenced[10)] by the ACPM.

For both the P-U-ABORT request and indication primitives: The User Data parameter is used to carry the ABRT APDU as specified below.

a) The APCI of the ABRT APDU is expressed using the ACSE abstract syntax of this Protocol Specification. This abstract syntax must be one of the available presentation contexts.

b) User information (if any) from the A-ABORT request primitive is expressed using one or more available presentation contexts.

8.3.2 *Use of P-P-ABORT indication parameter*

The Reason parameter of the provider-initiated P-P-ABORT indication primitive is mapped directly to the corresponding parameter of the A-P-ABORT indication.

8.4 *Association establishment (X.410-1984 mode)*

The association establishment procedure uses the P-CONNECT service.

10) If an association is supported by version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the User Data parameter is not referenced by the ACPM (because of length constraints) and is mapped directly onto the User Information parameter of the A-ABORT primitives (see § 7.3.3.1).

8.4.1 *Directly mapped parameters*

The following parameters are not referenced by the ACPM and are mapped directly onto corresponding parameters of the A-ASSOCIATE primitives:

a) User data[11];

b) Calling Presentation Address;

c) Called Presentation Address;

d) Responding Presentation Address;

e) Quality of Service;

f) ST 1 TEXT = Session Requirements;

g) Initial Synchronization Point Serial Number;

h) Initial Assignment of Tokens; and

i) Session-connection identifier.

8.4.2 *Use of other P-CONNECT request and indication parameters*

The Mode parameter of the P-CONNECT request and indication primitives is referenced by the ACPM.

For the P-CONNECT request primitive: The Mode parameter is set to the value of the Mode parameter of the A-ASSOCIATE request primitive. For the X.410-1984 mode of ACSE operation, this parameter has the value of "X.410-1984". This indicates to the presentation-service that it is to operate in the X.410-1984 mode for this presentation-connection.

For the P-CONNECT indication primitive: This parameter has the value of "X.410-1984" for the X.410-1984 mode of ACSE operation. This value indicates that the accepting ACPM is to operate in the X.410-1984 mode for this association. The Mode parameter of the A-ASSOCIATE indication primitive is set to the value of "X.410-1984."

8.4.3 *Use of other P-CONNECT response and confirm parameters*

The Result parameter of the P-CONNECT response and confirm primitives is used by the ACPM when operating in the X.410-1984 mode.

For the P-CONNECT response primitive: The value of the Result parameter is mapped from the Result parameter of the A-ASSOCIATE Result parameter as shown in Table 9/X.227.

For the P-CONNECT confirm primitive: The Result and Result Source parameters of the A-ASSOCIATE confirm primitive are mapped from the Result parameter as shown in Table 10/X.227.

TABLE 9/X.227

Mapping ACSE's result (X.410-1984 mode)

A-ASSOCIATE rsp [from]	P-CONNECT rsp [to]
accepted	acceptance
rejected (permanent)	user-rejection
rejected (transient)	user-rejection

11) User Data is mapped directly onto the A-ASSOCIATE User Information parameter. No explicit presentation context is available for it.

TABLE 10/X.227

Mapping Presentation's result (X.410-1984 mode)

P-CONNECTcnf [from]	A-ASSOCIATE cnf [to]	A-ASSOCIATE cnf [to]
acceptance	accepted	ACSE service user
user-rejection	rejected (permanent)	ACSE service-user
provider-rejection	rejected (permanent)	presentation service provider

8.5 *Normal release of an association (X.410-1984 mode)*

The normal release procedure uses the P-RELEASE service. The following parameters are not referenced by the ACPM and are mapped directly onto corresponding parameters of the A-RELEASE primitives:

a) Result; and

b) User Data.

8.6 *Abnormal release of an association (X.410-1984 mode)*

The abnormal release procedure uses the P-U-ABORT and P-P-ABORT services.

8.6.1 *Use of P-U-ABORT request and indication parameters*

For both the P-U-ABORT request and indication primitives: The User Data parameter is not referenced by the ACPM and is mapped directly onto the User Information parameter of the corresponding A-ABORT primitives.

8.6.2 *Use of P-P-ABORT indication parameter*

For the P-P-ABORT indication primitive: The Reason parameter is not referenced by the ACPM and is mapped directly onto the corresponding parameter of the A-P-ABORT indication primitive.

9 Structure and encoding of ACSE APDUs

9.1 *Abstract syntax of the ACSE APDUs*

The abstract syntax of each of the ACSE APDUs is specified in this clause using ASN.1 (CCITT Rec. X.208 | ISO 8824).

```
ACSE-1 { joint-iso-ccitt association-control(2) modules(0) apdus(0) version1(1) }
-- ACSE-1 refers to ACSE version 1

DEFINITIONS ::=

BEGIN

EXPORTS
        acse-as-id, ACSE-apdu,
        aCSE-id, Application-context-name,
        AP-Title, AE-qualifier,
        AE-title, AP-invocation-identifier
        AE-invocation-identifier,
        Mechanism-name, Authentication-value;
```

```
IMPORTS  Name, RelativeDistinguishedName
                FROM InformationFramework
                        { joint-iso-ccitt ds(5) modules(1) informationFramework(1)
                        } ;
-- The data types Name and RelativeDistinguishedName are imported from ISO 9594-2.
-- object identifier assignments

acse-as-id OBJECT IDENTIFIER ::=
  { joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0)
    version1(1)
  }
-- may be used to reference the abstract syntax of the ACSE APDUs

aCSE-id OBJECT IDENTIFIER ::=
  { joint-iso-ccitt association-control(2) as-id(3) acse-ase(1) version(1) }
-- may be used to identify the Association Control ASE.

-- top level CHOICE

ACSE-apdu  ::=  CHOICE
 { aarq  AARQ-apdu,
   aare  AARE-apdu,
   rlrq  RLRQ-apdu,
   rlre  RLRE-apdu,
   abrt  ABRT-apdu

 }

AARQ-apdu  ::=  [ APPLICATION 0 ]  IMPLICIT SEQUENCE
 { protocol-version                          [0]     IMPLICIT BIT STRING
                                                     { version1 (0) }
                                                     DEFAULT { version1 },
   application-context-name                  [1]     Application-context-name,
   called-AP-title                           [2]     AP-title
                                                     OPTIONAL,
   called-AE-qualifier                       [3]     AE-qualifier
                                                     OPTIONAL,
   called-AP-invocation-identifier           [4]     AP-invocation-identifier
                                                     OPTIONAL,
   called-AE-invocation-identifier [5]       AE-invocation-identifier
                                                     OPTIONAL,
   calling-AP-title                          [6]     AP-title
                                                     OPTIONAL,
   calling-AE-qualifier                      [7]     AE-qualifier
                                                     OPTIONAL,
   calling-AP-invocation-identifier          [8]     AP-invocation-identifier
                                                     OPTIONAL,
   calling-AE-invocation-identifier          [9]     AE-invocation-identifier OPTIONAL,
    -- The following field shall not be present if only the Kernel is used.
   sender-acse-requirements                  [10]    IMPLICIT ACSE-requirements
                                                     OPTIONAL,
    -- The following field shall only be present if the Authentication functional unit is selected.
    mechanism-name                           [11]    IMPLICIT Mechanism-name
                                                     OPTIONAL,
   -- The following field shall only be present if the Authentication functional unit is selected.
    calling-authentication-value             [12]    EXPLICIT Authentication-value OPTIONAL,
    implementation-information               [29]    IMPLICIT Implementation-data
                                                     OPTIONAL,
    user-information                         [30]    IMPLICIT Association-information
                                                     OPTIONAL

 }
```

```
AARE-apdu  ::=  [ APPLICATION 1 ]  IMPLICIT SEQUENCE
 { protocol-version                          [0]    IMPLICIT BIT STRING
                                                    { version1 (0) }
                                                    DEFAULT { version1 },
   application-context-name                  [1]    Application-context-name,
   result                                    [2]    Associate-result,
   result-source-diagnostic                  [3]    Associate-source-diagnostic,
   responding-AP-title                       [4]    AP-title
                                                    OPTIONAL,
   responding-AE-qualifier        [5]        AE-qualifier
                                                    OPTIONAL,
   responding-AP-invocation-identifier       [6]    AP-invocation-identifier
                                                    OPTIONAL,
   responding-AE-invocation-identifier       [7]    AE-invocation-identifier
                                                    OPTIONAL,
  -- The following field shall not be present if only the Kernel is used.
   responder-acse-requirements               [8]    IMPLICIT ACSE-requirements
                                                    OPTIONAL,
  -- The following field shall only be present if the Authentication functional unit is selected.
   mechanism-name                            [9]    IMPLICIT Mechanism-name
                                                    OPTIONAL,
  -- This following field shall only be present if the Authentication functional unit is selected.
   responding-authentication-value           [10]   EXPLICIT Authentication-value
                                                    OPTIONAL,
   implementation-information                [29]   IMPLICIT Implementation-data
                                                    OPTIONAL,
   user-information                          [30]   IMPLICIT Association-information
                                                    OPTIONAL
 }

RLRQ-apdu  ::=  [ APPLICATION 2 ] IMPLICIT SEQUENCE
 { reason                                    [0]    IMPLICIT Release-request-reason  OPTIONAL,
   user-information                          [30]   IMPLICIT Association-information OPTIONAL
 }

RLRE-apdu  ::=  [ APPLICATION 3 ] IMPLICIT SEQUENCE
 { reason                                    [0]    IMPLICIT Release-response-reason  OPTIONAL,
   user-information                          [30]   IMPLICIT Association-information  OPTIONAL
 }

ABRT-apdu ::= [ APPLICATION 4 ] IMPLICIT SEQUENCE
 { abort-source                              [0]    IMPLICIT ABRT-source,
    abort-diagnostic                         [1]    IMPLICIT ABRT-diagnostic OPTIONAL,
  -- This field shall not be present if only the Kernel is used.
   user-information                          [30]   IMPLICIT Association-information OPTIONAL
 }

ABRT-diagnostic ::= ENUMERATED
 { no-reason-given (1),
   protocol-error (2),
   authentication-mechanism-name-not-recognized (3),
   authentication-mechanism-name-required (4),
   authentication-failure (5),
   authentication-required (6)
 }

ABRT-source  ::=  INTEGER
 { acse-service-user              (0),
   acse-service-provider          (1)
 }

ACSE-requirements  ::=  BIT STRING
 { authentication (0) }

Application-context-name ::= OBJECT IDENTIFIER
-- ***** Application-entity title productions follow (not in alphabetical order)*****

AP-title ::= CHOICE { AP-title-form1, AP-title-form2 }
```

```
AE-qualifier ::= CHOICE { AE-qualifier-form1, AE-qualifier-form2 }
-- When both AP-title and AE-qualifier data values are present in an AARQ or AARE APDU, both must have the
-- same form to allow the construction of an AE-title as discussed in CCITT Rec. X.665 | ISO 9834-6.

AP-title-form1 ::= Name
-- The value assigned to AP-title-form1 is The Directory Name of an application-process title.

AE-qualifier-form1 ::= RelativeDistinguishedName
-- The value assigned to AE-qualifier-form1 is the relative distinguished name of a particular
-- application-entity of the application-process identified by AP-title-form1.

AP-title-form2 ::= OBJECT IDENTIFIER

AE-qualifier-form2 ::= INTEGER

AE-title ::= CHOICE { AE-title-form1, AE-title-form2 }
-- As defined in CCITT Rec. X.650 | ISO 7498-3, an application-entity title is composed of an application-process
-- title and an application-entity qualifier.  The ACSE protocol provides for the transfer of an application-entity title
-- value by the transfer of its component values.  However, the following data type is provided for
-- International Standards that reference a single syntactic structure for AE titles.

AE-title-form1 ::= Name
-- For access to The Directory (CCITT Rec. X.500-Series | ISO 9594), an AE title has AE-title-form1.  This value
-- can be constructed from AP-title-form1 and AE-qualifier-form1 values contained in an AARQ or AARE APDU.
-- A discussion of forming an AE-title-form1 from AP-title-form1 and AE-qualifier-form1 may be found in
-- CCITT Rec. X.665 | ISO 9834-6.

AE-title-form2 ::= OBJECT IDENTIFIER
-- A discussion of forming an AE-title-form2 from AP-title-form2 and AE-qualifier-form2 may be found in
-- CCITT Rec. X.665 | ISO 9834-6.

AE-invocation-identifier ::= INTEGER

AP-invocation-identifier ::= INTEGER
-- ***** End of Application-entity title productions*****

Associate-result  ::= INTEGER
 { accepted (0),
   rejected-permanent (1),
   rejected-transient (2)
 }

Associate-source-diagnostic  ::=  CHOICE
 { acse-service-user [1] INTEGER
   { null (0),
     no-reason-given (1),
     application-context-name-not-supported (2),
     calling-AP-title-not-recognized (3),
     calling-AP-invocation-identifier-not-recognized (4),
     calling-AE-qualifier-not-recognized (5),
     calling-AE-invocation-identifier-not-recognized (6),
     called-AP-title-not-recognized (7),
     called-AP-invocation-identifier-not-recognized (8),
     called-AE-qualifier-not-recognized (9),
     called-AE-invocation-identifier-not-recognized (10),
     authentication-mechanism-name-not-recognized (11),
     authentication-mechanism-name-required (12),
     authentication-failure (13),
     authentication-required (14)
   },
   acse-service-provider [2] INTEGER
   { null (0),
     no-reason-given (1),
     no-common-acse-version (2)
   }
 }
```

```
Association-information ::=  SEQUENCE OF EXTERNAL

Authentication-value  ::=  CHOICE
 { charstring        [0]    IMPLICIT GraphicString,
   bitstring         [1]    IMPLICIT BIT STRING,
   external          [2]    IMPLICIT EXTERNAL,
   other             [3]    IMPLICIT SEQUENCE {
                                  other-mechanism-name        Mechanism-name,
                                  other-mechanism-value       ANY DEFINED BY
                                                                other-mechanism-name
                                                              }
 }
-- The abstract syntax of authentication-value is determined by the authentication-mechanism used during
-- association stablishment.  The authentication-mechanism is either explicitly denoted by the
-- OBJECT IDENTIFIER value for mechanism-name, or it is known implicitly by prior agreement between the
-- communicating partners.  If "other" is chosen, then "mechanism-name" must be present in accordance with
-- CCITT Rec. X.208 | ISO 8824. If the value "mechanism-name" occurs in the AARQ-apdu or the AARE-apdu,
-- then that value must be the same as the value for "other-mechanism-name"

Implementation-data ::=  GraphicString

Mechanism-name  ::=  OBJECT IDENTIFIER
-- This field shall be present if authentication-value is of type ANY DEFINED BY.

Release-request-reason  ::=  INTEGER
 { normal            (0),
   urgent            (1),
   user-defined      (30)
 }

Release-response-reason  ::=  INTEGER
 { normal            (0),
   not-finished      (1),
   user-defined      (30)
 }

END
```

9.2 *Encoding rules used to reference the transfer syntax*

The set of encoding rules named

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

and specified in CCITT Rec. X.209 | ISO 8825 is used to reference the transfer syntax of the ACSE APDUs in the context of the ACSE abstract syntax definition.

10 Conformance

A system claiming to implement the procedures specified in this Protocol Specification shall comply with the requirements in § 10.1 through § 10.3.

Two modes of conformance are recognized:

a) normal mode; and

b) X.410-1984 mode.

The X.410-1984 mode exists to allow claims of conformance to allow compatibility with message handling systems implementing the protocol specified in CCITT Recommendation X.410-1984.

10.1 *Statement requirements*

The following shall be stated by the implementor:

a) whether the system is capable of acting in the role of association-initiator, or association-responder, or both;

b) that the system supports version 1 of this protocol; and

c) whether the system implements:

1) the normal mode of ACSE protocol;

2) the X.410-1984 mode of ACSE protocol to support a message handling system; or

3) both the normal mode and the X.410-1984 mode for the reason given in item 2) above.

10.2 *Static requirements*

The use of the Association Control Service Element is required for an application-entity to meet the minimum requirements for establishing and releasing communication with a peer entity.

10.2.1 *Normal mode*

If the normal mode is implemented, the system shall:

a) act in the role of an association-initiator (by sending an AARQ APDU), or in the role of an association-acceptor (by responding properly to an AARQ APDU with an appropriate AARE APDU), or act in both roles; and

b) support (as a minimum) that encoding which results from applying the basic ASN.1 encoding rules to the ASN.1 specified in § 9 for the purpose of exchanging ACSE APCI.

10.2.2 *X.410-1984 mode*

If the X.410-1984 mode is implemented, the system shall act in the role of an initiator, or acceptor, or both.

10.3 *Dynamic requirements*

10.3.1 *Normal mode*

If the normal mode is implemented, the system shall:

a) follow all the procedures specified in § 7 (including the rules for extensibility) and Annex A; and

b) support the mapping onto the Presentation Service defined in § 8.1 to § 8.3.

The requesting AE may choose to send either form 1 or form 2 for AP title and AE qualifier. The accepting AE may respond with either form. Thus, both the requesting and responding AE must be capable of receiving both forms.

10.3.2 *X.410-1984 mode*

If the X.410-1984 mode is implemented, the system shall support the direct mapping of parameters of presentation-service primitives onto the ACSE primitives as specified in § 8.4 to § 8.6.

11 Precedence

The aspects of the protocol for ACSE are specified in several clauses in this Protocol Specification. This clause states the rules of precedence for possible situations where the same aspect may be specified in more than one place in an apparently inconsistent manner. The relevant aspects of protocol specification covered are:

a) sequencing rules;

b) mapping to the presentation-service; and

c) structure and encoding of ACSE APDUs.

Annex A and § 7 of this International Standard specify the elements of procedure which govern the behavior of the ACPM. Annex A takes precedence over any other clause in this Protocol Specification which may state or imply apparently inconsistent sequencing rules.

How the presentation service primitives are used by the ACPM is specified in § 8 which takes precedence over any other part of this Protocol Specification which may state or imply mapping to the presentation-service.

The structure and encoding of ACSE APDUs are specified in § 9 which takes precedence over any other part of this Protocol Specification which may state or imply structure and encoding of ACSE APDUs.

Note – Any person encountering an inaccuracy or ambiguity in this Protocol Specification is requested to notify their National Body of ISO without delay in order that the matter may be investigated and appropriate action taken.

12 Registration requirements

This Protocol Specification identifies the requirement to register three types of information objects: application titles; application contexts; and authentication-mechanisms. Each is discussed below.

No International Registration Authority is currently planed for the registration of any of the above objects. The assignment of a name to any of these objects shall be in accordance with the general provisions of CCITT Rec. X.660 | ISO/IEC 9834-1, except as specified below.

In accordance with CCITT Rec. X.660 | ISO/IEC 9834-1, an organization that wishes to assign names to objects shall find an appropriate superior of the naming tree. The superior assigns an arc of the naming tree to the organization. The organization can then assign names below that arc.

Note – Appropriate superiors in the registration tree include ISO/IEC national bodies, organizations with International Code Designators assigned in accordance with ISO 6523, as well as CCITT network administrations, and recognized private operating agencies (RPOAs).

12.1 *Application titles*

The application titles requiring registration are application-process title, application-entity qualifier, and application-entity title. The registration requirements for these information objects are contained in ISO/IEC 9545, § 9. CCITT Rec. X.665 | ISO/IEC 9834-6 specifies both the relationship between these information objects and the procedures to register them.

12.2 *Application context*

The registration requirements for an application context is contained in ISO/IEC 9545, § 9. CCITT Rec. X.660 | ISO/IEC 9834-1 specifies the procedures to register it.

12.3 *Authentication-mechanism*

An authentication-mechanism may be specified as part of a CCITT Recommendation or ISO/IEC International Standard. For example, Annex B of this Protocol Specification includes an authentication-mechanism based on AE title and password. Such an authentication-mechanism is, in effect, specified and registered within this Protocol Specification.

An authentication-mechanism may also be specified outside of CCITT Recommendations and ISO/IEC International Standards. In this situation, CCITT Rec. X. 660 | ISO/IEC 9834-1 specifies the procedures to register such an authentication-mechanism.

ANNEX A

ACPM state table

(This annex forms an integral part of this Recommendation)

A.1 *General*

This annex defines a single Association Control Protocol Machine (ACPM) in terms of a state table. The state table shows the interrelationship between the state of an ACPM, the incoming events that occur in the protocol, the actions taken, and, finally, the resultant state of the ACPM.

The ACPM state table does not constitute a formal definition of the ACPM. It is included to provide a more precise specification of the elements of procedure defined in § 7.

This annex contains the following tables:

a) Table A.1/X.227 specifies the abbreviated name, source, and name/description of each incoming event. The sources are:

   1) ACSE service-user (AC-user);

   2) Peer ACPM (AC-peer); and

   3) presentation service-provider (PS-provider).

b) Table A.2/X.227 specifies the abbreviated name of each state.

c) Table A.3/X.227 specifies the abbreviated name, target, and name/description of each outgoing event. The targets are:

   1) ACSE service-user (AC-user); and

   2) peer ACPM (AC-peer).

d) Table A.4/X.227 specifies the predicates.

e) Table A.5/X.227 specifies the ACPM state table using the abbreviations of the above tables.

A.2 *Conventions*

The intersection of an incoming event (row) and a state (column) forms a cell.

In the state table, a blank cell represents the combination of an incoming event and a state that is not defined for the ACPM (see § A.3.1).

A non-blank cell represents an incoming event and state that is defined for the ACPM. Such a cell contains one or more action lists. An action list may be either mandatory or conditional. If a cell contains a mandatory action list, it is the only action list in the cell.

A mandatory action list contains

a) an outgoing event; and

b) a resultant state.

A conditional action list contains

a) a predicate expression comprising predicates and Boolean operators (^ represents the Boolean NOT); and

b) a mandatory action list. (This mandatory action list is used only if the predicate expression is true.)

A.3 *Actions to be taken by the ACPM*

The ACPM state table defines the action to be taken by the ACPM in terms of an outgoing event and the resultant state of the ACPM.

TABLE A-1/X.227

Incoming event list

Abbreviated name	Source	Name and description
A-ASCreq	AC-user	A-ASSOCIATE request primitive
A-ASCrsp+	AC-user	A-ASSOCIATE response primitive (Result = "accepted")
A-ASCrsp–	AC-user	A-ASSOCIATE response primitive [Result = "rejected (permanent)" or "rejected (transient)"]
AARQ	AC-peer	A-ASSOCIATE-REQUEST-APDU The AARQ is user data on a P-CONNECT indication
AARE+	AC-peer	A-ASSOCIATE-RESPONSE-APDU (Result = "accepted") The AARE+ is user data on a P-CONNECT confirm primitive (Result = "acceptance")
AARE–	AC-peer	A-ASSOCIATE-RESPONSE APDU [Result = "rejected (permanent)" or "rejected (transient)"] The AARE– is user data on a P-CONNECT confirm primitive (Result = "user-rejection")
P-CONcnf–	PS-provider	P-CONNECT confirm primitive (Result = "provider-rejection")
A-RLSreq	AC-user	A-RELEASE request primitive
A-RLSrsp+	AC-user	A-RELEASE response primitive (Result = "affirmative")
A-RLSrsp–	AC-user	response primitive A-RELEASE response primitive (Result = "negative")
RLRQ	AC-peer	A-RELEASE-REQUEST APDU The RLRQ is user data on a P-RELEASE indication primitive
RLRE+	AC-peer	A-RELEASE-RESPONSE APDU The RLRE+ is user data on a P-RELEASE confirm primitive (Result = "affirmative")
RLRE–	AC-peer	A-RELEASE-RESPONSE APDU The RLRE– is user data on a P-RELEASE confirm primitive (Result = "negative")
A-ABRreq	AC-user	A-ABORT request primitive
ABRT[a)]	AC-peer	A-ABORT APDU The ABRT is user data on a P-U-ABORT indication primitive
P-PABind	PS-provider	P-P-ABORT indication primitive

a) When supported by version 1 of the version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the A-ABORT APDU has no APCI. The receipt of the P-U-ABORT indication implies its existence.

A.3.1 *Invalid intersections*

Blank cells indicate an invalid intersection of an incoming event and state. If such an intersection occurs, one of the following actions is taken:

a) if the incoming event comes from the ACSE service-user, any action taken by the ACPM is a local matter;

b) if the incoming event is related to a received APDU or PS-provider event, the ACPM issues both an A-ABRind outgoing event (to its AC-user) and an ABRT outgoing event (to its peer ACPM).

A.3.2 *Valid intersections*

If the intersection of the state and incoming event is valid, one of the following actions is taken:

a) if a cell contains a mandatory action list, the ACPM takes the actions specified;

b) if a cell contains one or more conditional action lists, for each predicate expression that is true, the ACPM takes the actions specified. If none of the predicate expressions are true, the ACPM takes one of the actions defined in § A.3.1.

A.4 *Relationship to Presentation and other ASEs*

The ACPM state table (Table A-5/X.227) only defines the interactions of the ACPM, its ACSE service-user and the presentation-services used by the ACPM.

Note – The occurrence of other events from the presentation-service or other application-service-elements is not included in the ACPM state table because they do not affect the ACPM.

TABLE A-2/X.227

ACPM states

Abbreviated name	Description
STA0	idle: unassociated
STA1	awaiting AARE APDU
STA2	awaiting A-ASSOCIATE response
STA3	awaiting RLRE APDU
STA4	awaiting A-RELEASE response
STA5	associated
STA6	awaiting A-RELEASE response (association-initiator)
STA7	awaiting RLRE APDU (association-responder)

TABLE A-3/X.227

Outgoing event list

Abbreviated name	Target	Name and description
A-ASCind	AC-user	A-ASSOCIATE indication primitive
A-ASCcnf+	AC-user	A-ASSOCIATE confirm primitive (Result = "accepted")
A-ASCcnf–	AC-user	A-ASSOCIATE confirm primitive (Result = "rejected (permanent)" or "rejected (transient)")
AARQ	AC-peer	A-ASSOCIATE-REQUEST APDU The AARQ is sent as user data on a P-CONNECT request primitive
AARE+	AC-peer	A-ASSOCIATE-RESPONSE APDU (Result = "accepted") The AARE+ is sent as user data on a P-CONNECT+ response primitive (Result = "acceptance")
AARE–	AC-peer	A-ASSOCIATE-RESPONSE APDU (Result = "rejected (permanent)" or "rejected (transient)") The AARE– is sent as user data on a P-CONNECT– response primitive (Result = "user-rejection")
A-RLSind	AC-user	A-RELEASE indication primitive
A-RLScnf+	AC-user	A-RELEASE confirm primitive (Result = "affirmative")
A-RLScnf–	AC-user	A-RELEASE confirm primitive (Result = "negative")
RLRQ	AC-peer	A-RELEASE-REQUEST ADPU The RLRQ is sent as user data on a P-RELEASE request primitive
RLRE+	AC-peer	A-RELEASE-RESPONSE APDU The RLRE+ is sent as user data on a P-RELEASE response primitive (Result = "affirmative")
RLRE–	AC-peer	A-RELEASE-RESPONSE APDU The RLRE– is sent as user data on a P-RELEASE response primitive (Result = "negative")
A-ABRind	AC-user	A-ABORT indication primitive (Source = "ACSE service user" or "ACSE service-provider")
ABRT[a)]	AC-peer	A-ABORT APDU (Source = "ACSE service-user" or "ACSE service-provider") The ABRT is sent as user data on a P-U-ABORT request primitive
P-PABind	AC-user	A-P-ABORT indication primitive

a) When supported by version 1 of the version 1 of the session-protocol (CCITT Rec. X.225 | ISO 8327), the A-ABORT APDU has no APCI. The receipt of the P-U-ABORT indication implies its existence.

TABLE A-4/X.227

Predicates

Code	Meaning
p1 p2	ACPM can support requested connection ACPM originated this association

TABLE A-5/X.227

ACPM state table

	STA0 Idle-Unassoc	STA1 Awaiting AARE	STA2 Awaiting A-ASCrsp	STA3 Awaiting RLRE	STA4 Awaiting A-RLSrsp	STA5 Associated	STA6 Collision association initiator	STA7 Collision association responder
A-ASCreq	p1 AARQ STA1							
A-ASCrsp+			AARE+ STA5					
A-ASCrsp–			AARE– STA0					
AARQ	p1 A-ASCind STA2 ^p1: AARE– STA0							
AARE+		A-ASCcnf+ STA5						
AARE–		A-ASCcnf– STA0						
P-CONcnf–		A-ASCcnf– STA0						
A-RLSreq						RLRQ STA3		
A-RLSrsp+					RLRE+ STA0		RLRE+ STA3	
A-RLSrsp–					RLRE– STA5			
RLRQ				p2 A-RLSind STA6 ^p2 A-RLSind STA7		A-RLSind STA4		
RLRE+				A-RLScnf+ STA0				A-RLScnf+ STA4
RLRE–				A-RLScnf– STA5				
A-ABRreq		ABRT STA0	ABRT STA0	ABRT STA0	ABRT STA0	ABRT STA0	ABRT STA0	ABRT STA0
ABRT		A-ABRind STA0	A-ABRind STA0	A-ABRind STA0	A-ABRind STA0	A-ABRind STA0	A-ABRind STA0	A-ABRind STA0
P-PABind		A-PABind STA0	A-PABind STA0	A-PABind STA0	A-PABind STA0	A-PABind STA0	A-PABind STA0	A-PABind STA0

ANNEX B

Authentication-mechanism using password

(This annex forms an integral part of this Recommendation)[12)]

B.0 *Introduction*

This annex specifies a simple authentication-mechanism that uses a password with an AE title. This authentication-mechanism is intended for general use. It is also an example of an authentication-mechanism specification.

B.1 *Assigned name*

The following name (of ASN.1 datatype OBJECT IDENTIFIER) is assigned to this authentication-mechanism:

{ joint-iso-ccitt association-control(2) authentication-mechanism(3) password-1(1) }

B.2 *Authentication-value ASN.1 datatype*

For this authentication-mechanism, the password is the authentication-value. The data type of authentication-value shall be "GraphicString" in accordance with the production for "authentication-value" in § 9.

B.3 *Processing specification*

In this annex, the term "sending" denotes the AEI (or its authentication-function) requesting authentication by its peer. The term "receiving" denotes the AEI (or its authentication-function) performing authentication of its peer.

B.3.1 *Requesting authentication*

The sending authentication-function retrieves a password value from stored data for its AEI to be corroborated by the receiving AEI. The password value is mapped to the datatype of the authentication-value defined in § B.2.

When the A-ASSOCIATE request or response primitive is issued by the sending AEI, the Authentication-value parameter shall contain this value. The primitive shall also contain the appropriate AP title and AE qualifier parameters that indicates its AE title.

Depending on prior agreements between the sending and the receiving AE, the authentication-mechanism name (defined in § B.1) may or may not be included on the A-ASSOCIATE primitive.

B.3.2 *Performing authentication*

The receiving authentication-function receives the Authentication-value parameter value on the incoming A-ASSOCIATE indication or confirm primitive.

Depending on prior agreements between the sending and the receiving AE, the authentication-mechanism name (defined in § B.1) may or may not be included on the A-ASSOCIATE primitive.

If an authentication-mechanism name is required but not received, the authentication-function indicates that an A-ABORT request primitive shall be issued. The Diagnostic parameter value shall indicate "authentication-mechanism name required."

---

12) The implementation of the authentication-mechanism specified in this annex is not required for conformance to this Protocol Specification. However, if employed, the entire Specification is binding for the authentication-functions in both AEIs.

If an authentication-mechanism name is included, it shall be semantically equivalent to that specified in § B.1. If this authentication-mechanism name is not correct, the authentication-function indicates that an A-ABORT request primitive shall be issued. The Diagnostic parameter value shall indicate "authentication-mechanism name not recognized."

The authentication-function then determines if this authentication-mechanism (i.e., the authentication-mechanism defined in this annex) is allowed for the sending AEI based on the AE title of the sending AEI. If this authentication-mechanism is not allowed, the authentication-function indicates that an A-ABORT request primitive shall be issued. The Diagnostic parameter value shall indicate "authentication failure."

If this authentication-mechanism is allowed for the sending AEI, the authentication-function compares the value of the Authentication-value parameter against its stored data for this mechanism based on the sender's AE title. If the two are semantically equivalent, the authentication-function shall indicate successful authentication.

If the two values are not semantically equivalent, the authentication-function indicates that an A-ABORT request primitive shall be issued. The diagnostic parameter value shall indicate "authentication failure."

Printed in Switzerland

Geneva, 1993