

Recommendation X.227

ASSOCIATION CONTROL PROTOCOL SPECIFICATION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS ¹

(Melbourne, 1988)

The CCITT,

considering

- (a) that Recommendation X.200 defines the Reference Model of Open Systems Interconnection for CCITT applications;
- (b) that Recommendation X.208 specifies Abstract Syntax Notation One (ASN.1) for the specification of the abstract syntax of protocols;
- (c) that Recommendation X.209 specifies the basic encoding rules for Abstract Syntax Notation One;
- (d) that Recommendation X.210 defines the Open Systems Interconnection (OSI) layer service definition conventions;
- (e) that Recommendation X.215 defines the Session service definition for Open Systems Interconnection for CCITT applications;
- (f) that Recommendation X.216 defines the Presentation service definition of Open Systems Interconnection for CCITT applications;
- (g) that Recommendation X.217 defines Association Control service definition for Open Systems Interconnection for CCITT applications;
- (h) that Recommendation X.220 specifies the use of X.200 series protocols in CCITT applications;
- (i) that Recommendation X.410-1984 specifies the protocol for Remote Operation and Reliable Transfer Server for Message Handling Systems; and
- (j) that there is a need for common Association Control support for various applications,

unanimously declares

that this Recommendation defines the Association Control specification of Open Systems Interconnection for CCITT applications as given in the Scope and Field of Application.

CONTENTS

0	Introduction
1	Scope and field of application
2	References
3	Definitions
3.1	Reference Model definitions
3.2	Naming and addressing definitions
3.3	Service conventions definitions

¹Recommendation X.227 and ISO 8650 [Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element] were developed in close collaboration and are technically aligned, except for the differences noted in Appendix I.

	3.4	Presentation service definitions
	3.5	ACSE service definitions
	3.6	Association Control protocol specification definitions
4		Symbols and abbreviations
	4.1	Data units
	4.2	Types of application-protocol-data-units
	4.3	Other abbreviations
5		Conventions
6		Overview of the protocol
	6.1	Service provision
	6.2	Use of the presentation-service
	6.3	Relationship to the session-service
	6.4	Model
7		Elements of procedure
	7.1	Association establishment
	7.2	Normal release of an association
	7.3	Abnormal release of an association
	7.4	Rules for extensibility
8		Mapping to the presentation-service
	8.1	Association establishment (normal mode)
	8.2	Normal release of an association (normal mode)
	8.3	Abnormal release of an association (normal mode)
	8.4	Association establishment (X.410-1984 mode)
	8.5	Normal release of an association (X.410-1984 mode)
	8.6	Abnormal release of an association (X.410-1984 mode)
9		Structure and encoding of ACSE APDUs
10		Conformance
	10.1	Statement requirements
	10.2	Static requirements
	10.3	Dynamic requirements

Annex A - ACPM state table for normal mode of operation

A.1	General
A.2	Conventions
A.3	Actions to be taken by the ACPM
	A.3.1 Invalid intersections
	A.3.2 Valid intersections
A.4	Relationship to Presentation and other ASEs

Annex B - ACPM state table for X.410-1984 mode of operation

B.1 General

B.2 Conventions

B.3 Actions to be taken by the ACPM

B.3.1 Invalid intersections

B.3.2 Valid intersections

B.4 Relationship to Presentation and other ASEs

Appendix I - Differences between Recommendation X.227 and ISO International Standard 8650

Appendix II - Summary of Assigned Object Identifier Values

0 Introduction

0.1 This Recommendation is one of a set of Recommendations produced to facilitate the interconnection of information processing systems. It is related to other Recommendations in the set as defined by the Reference Model for Open Systems Interconnection (X.200). The Reference Model subdivides the area of standardization for interconnection into a series of layers of specification, each of manageable size.

0.2 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.

0.3 This Recommendation specifies the protocol for the application-service-element for application-association control: the Association Control Service Element (ACSE). The ACSE provides services for establishing and releasing application-associations. These services are intended to be applicable to a wide range of application-process communication requirements.

0.4 This Recommendation includes two annexes which describe the protocol machine of ACSE in terms of a state table for normal mode of operation and for X.410-1984 mode of operation. This protocol machine is referred to as the Association Control Protocol Machine (ACPM).

0.5 The protocol defined in this Recommendation is also governed by the use of the presentation-service (X.216) and the session-service (X.215).

0.6 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 individual treatments in each layer service satisfy overall QOS objectives in a consistent manner. As a consequence, a change may be made to this Recommendation at a later time which reflects further QOS developments and integration.

1 Scope and field of application

The procedures defined in this Recommendation are applicable to instances of communication between systems which wish to interconnect in an open systems interconnection environment. This Recommendation specifies:

- a) procedures for the transfer of information relating to the application-association control between 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 Recommendation also specifies conformance requirements for systems implementing these procedures. It does not contain tests which can be used to demonstrate conformance.

2 References

- Recommendation X.200 - Reference Model of Open Systems Interconnection for CCITT applications (see also ISO 7498-1).
- Recommendation X.208 - Specification of Abstract Syntax Notation One (see also ISO 8824).
- Recommendation X.209 - Basic Encoding Rules for Abstract Syntax Notation One (see also ISO 8825).
- Recommendation X.210 - OSI Layer Service Definition Conventions (see also ISO TR 8509).
- Recommendation X.215 - Session service definition for Open Systems Interconnection for CCITT applications (see also ISO 8326 and ISO 8326 Addendum 2).
- Recommendation X.216 - Presentation service definition for Open Systems Interconnection for CCITT applications (see also ISO 8822).
- Recommendation X.217 - Association Control service definition for Open Systems Interconnection for CCITT applications (see also ISO 8649).
- Recommendation X.225 - Session protocol specification for Open Systems Interconnection for CCITT applications (see also ISO 8327 and ISO 8327 Addendum 2).
- Recommendation X.410 - CCITT Recommendation X.410: Message Handling Systems: Remote Operations and Reliable Transfer Server (1984).
- ISO 7498-3 - Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and Addressing.

3 Definitions

3.1 *Reference Model definitions*

This Recommendation is based on the concepts developed in X.200 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-service protocol; and
- k) session-connection.

3.2 *Naming and addressing definitions*

This Recommendation makes use of the following terms defined in ISO 7498-3:

- a) application-process title;
- b) application-entity qualifier;
- c) application-entity title²
- d) application-process invocation-identifier;
- e) application-entity invocation-identifier; and
- f) presentation address.

3.3 *Service conventions definitions*

This Recommendation makes use of the following terms defined in X.210:

- 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).

3.4 *Presentation service definitions*

This Recommendation makes use of the following terms defined in X.216:

- 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].

²As defined in 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.

3.5 *ACSE service definitions*

This Recommendation makes use of the following terms defined in X.217.

- a) application-association; association;
- b) application context;
- c) Association Control Service Element;
- d) ACSE service-user;
- e) ACSE service-provider;
- f) requestor;
- g) acceptor;
- h) association-initiator;
- i) association-responder;
- j) normal mode;
- k) X.410-1984 mode; and
- j) disrupt.

3.6 *Association Control protocol specification definitions*

The following terms are introduced in this Recommendation.

3.6.1 **Association Control Protocol Machine**

The protocol machine for the Association Control Service Element specified in this Recommendation.

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 **Symbols and 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 Recommendation.

AARQ A-ASSOCIATE-REQUEST application-protocol-data-unit

AARE A-ASSOCIATE-REQUEST application-protocol-data-unit

RLRQ A-RELEASE-REQUEST application-protocol-data-unit

RLRE A-RELEASE-RESPONSE application-protocol-data-unit
ABRT A-ABORT application-protocol-data-unit

4.3 *Other abbreviations*

The following abbreviations are used in this Recommendation:

ACPM Association Control Protocol Machine
ACSE Association Control Service Element
AE application-entity
AP application-process
APCI application-protocol-control-information
ASE application-service-element
ASN.1 Abstract Syntax Notation One
OSI Open Systems Interconnection
QOS quality of service

5 **Conventions**

5.1 This Recommendation 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
sp source or sink is the ACPM

5.2 The structure of each ACSE SPDU is specified in § 9 using the abstract syntax notation of ASN.1 (X.208).

6 **Overview of the protocol**

6.1 *Service provision*

The protocol specified in this Recommendation provides the services defined in X.217. These services are listed in Table 1/X.227. For a particular association, the ACSE 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

Service summary

Service	Type
A-ASSOCIATE	Confirmed
A-RELEASE	Confirmed
A-ABORT	Non-confirmed
A-P-ABORT	Provider-initiated

6.2 *Use of the presentation-service*

6.2.1 **ACE's use of the presentation-service (X.216) 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. 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.

6.2.2 This Recommendation 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.

6.2.3 When supported by version 1 of the session-protocol (X.225), the presentation-service is subject to length restrictions for its user-data parameters. This Recommendation 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).

6.3 *Relationship to the session-service*

6.3.1 The functional units of the session-service (X.215) required for the session-connection which support 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.

6.3.2 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 Recommendation assumes that a local mechanism enforces them. Some examples of session-service constraints which affect the ACSE service-user are:

- a) the availability of negotiated release; and
- b) the possibility of release collisions.

6.4 *Model*

6.4.1 The Association control Protocol Machine (ACPM) is modeled as a finite state machine whose specification is given in this Recommendation. The ACPM communicates with its service-user by means of the ACSE service primitives defined in X.217. The ACPM communicates with its presentation service-provider by means of the presentation services defined in X.216.

6.4.2 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 which 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.

6.4.3 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.

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

6.4.5 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 Recommendation.

6.4.6 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.

6.4.7 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 § 9³. 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 which is used. The use of the parameters of the presentation primitives are described in § 8.

A detailed specification of the ACSE APDUs using the ASN.1 notation (X.208) is described in § 9. Annex A specifies the state table for the ACPM for normal mode of operation. Annex B specifies the state table for the ACPM for X.410-1984 mode of operation.

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.

³ This is true with one exception. If the association is supported by version 1 of the session-protocol (X.225), 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 §§ 6.4.6 and 7.3.3.1).

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 PDU are listed in Table 2/X.227. The fields of the AARE APDU are listed in Table 3/X.227.

TABLE 2/X.227

AARQ APDU fields

Field name	Presence	Source	Sink
Protocol Version	O	sp	sp
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
Implementation information	O	sp	sp
User information	U	req	ind

TABLE 3/X.227

AARE APDU fields

Field name	Presence	Source	Sink
Protocol Version	O	sp	sp
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/sp	cnf
Result Source - Diagnostic	M	rsp/sp	cnf
Implementation Information	O	sp	sp
User Information	U	rsp	cnf

7.1.3 Association establishment procedure

This procedure is driven by the following events:

- an A-ASSOCIATE request primitive from the requestor;
- an AARQ APDU as user data on a P-CONNECT indication primitive;
- an A-ASSOCIATE response primitive from the acceptor; and
- a P-CONNECT confirm primitive (that may or may not contain an AARE APDU).

7.1.3.1 A-ASSOCIATE request primitive

7.1.3.1.1 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.

7.1.3.1.2 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

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

7.1.3.2.2 The ACPM determines if the AARQ ADPU 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.

7.1.3.2.3 The ACPM next inspects the value of the Protocol Version field ⁴of 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 which indicates the protocol version(s) which 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
- d) Result Source-Diagnostic field with the values »ACSE service-provider« and »not 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 which has the value »user rejection«. The ACPM does not issue an A-ASSOCIATE indication primitive. The association is not established.

7.1.3.2.4 If the P-CONNECT indication primitive and its AARQ APDU are acceptable, the ACPM 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

7.1.3.3.1 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.

7.1.3.3.2 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.

7.1.3.3.3 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.

7.1.3.4 P-CONNECT confirm primitive

7.1.3.4.1 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 representation service-provider has rejected the related presentation connection.

7.1.3.4.2 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.

7.1.3.4.3 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.

7.1.3.4.4 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.

4 If the Protocol Version field is not present in the AARQ APDU, version 1 is assumed

7.1.3.4.5 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«⁵ 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 Recommendation which 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 which are higher than the one indicating the latest version of this Recommendation which 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.

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.

⁵The presentation-service (Rec. X.216) 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, a change may be made to this Recommendation at a later time that reflects further developments and integration.

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 determined by the value of the Called AE Invocation-identifier parameter of the A-ASSOCIATE indication primitive, if issued.

7.1.3.11 *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.

7.1.4.12 *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 which 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 which indicates the protocol version(s) of this Recommendation which 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 Recommendation 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 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 *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.

7.1.5.10 *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 § 6.4.6). For a given AE, two distinct ACPMs would be involved which represent the processing for two distinct associations:

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

- b) an ACPM which 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 4/X.227. The fields of the RLRE APDU are listed in Table 5/X.227.

TABLE 4/X.227

RLRQ APDU fields

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

TABLE 5/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 P-RELEASE confirm primitive.

7.2.3.1 *A-RELEASE request primitive*

7.2.3.1.1 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 § 6.2 and 6.3).

7.2.3.1.2 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 § 6.3).

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*

7.2.3.5.1 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.

7.2.3.5.2 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 which 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 an 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 which 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 6/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 6/X.227

ABRT APDU fields

Field name	Presence	Source	Sink
Abort Source	M	sp	ind
User Information	U	req	ind

7.3.3 Abnormal release procedure

This procedure is driven by the following events:

- an A-ABORT request primitive from the requestor;
- 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 which it performs depends on the version of the underlying session-protocol (X.225) which 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 the 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 contain⁶ 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 a 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 ADPU, 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*

7.3.3.4.1 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).

7.3.3.4.2 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.

7.3.3.4.3 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 procedure⁷ 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.

7.3.3.4.4 The subsequent ACPM processing performed depends on the version of the underlying session-protocol (X.225) which supports the association as specified below.

⁶If an association is supported by version 1 of the session-protocol (Rec. X.225), the User Data parameter does not contain an ABRT ADPU (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.

⁷Since an A-ASSOCIATE indication primitive is not issued, an A-ABORT indication primitive would have no meaning, and, therefore, it is not issued.

- 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.

7.3.3.4.5 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 *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 (X.216).

7.4 *Rules for extensibility*

7.4.1 When processing an incoming AARQ, the accepting ACPM shall:

- a) ignore all tagged values which are not defined in the abstract syntax of this Recommendation; and
- b) ignore all unknown bit name assignments within a bit string.

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

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