

INTERNATIONAL STANDARD

IEC
62056-53

Second edition
2006-12

**Electricity metering –
Data exchange for meter reading,
tariff and load control –**

**Part 53:
COSEM application layer**



Reference number
IEC 62056-53:2006(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site** (www.iec.ch)
- **Catalogue of IEC publications**
The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.
- **IEC Just Published**
This summary of recently issued publications (www.iec.ch/online_news/justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.
- **Customer Service Centre**
If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 62056-53

Second edition
2006-12

Electricity metering – Data exchange for meter reading, tariff and load control –

Part 53: COSEM application layer

© IEC 2006 — Copyright - all rights reserved

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission, 3, rue de Varembe, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE **XG**

For price, see current catalogue

CONTENTS

FOREWORD.....	5
1 Scope.....	7
2 Normative references	7
3 Terms, definitions and abbreviations	8
4 The COSEM communications framework	10
4.1 Client/server type operation, communication profiles	10
4.2 Connection (association) oriented operation	12
5 Overview: the COSEM application layer	12
5.1 Specification method	12
5.2 Application layer structure	12
5.3 Service specification	13
5.4 Layer management services	15
5.5 Protocol specification	15
6 COSEM application layer – Service specification	16
6.1 Summary of services	16
6.2 Application association establishment and release	16
6.3 Special application associations	17
6.4 Data communication	18
6.5 Client COSEM application layer services	19
6.6 Server COSEM application layer services	38
6.7 Summary of COSEM application layer services and service parameters	55
7 COSEM application layer protocol specification	59
7.1 State definitions for the client side control function	59
7.2 State definitions for the server side control function	61
7.3 Protocol for application association establishment/release	62
7.4 Protocol for data communications	74
8 Specification of COSEM data types	89
8.1 The COSEM APDUs	89
8.2 The ACSE APDUs	90
8.3 Useful types	93
8.4 The xDLMS-Initiate.request/response/ConfirmedServiceError PDUs	98
8.5 The conformance block	98
8.6 Definition of APDUs for data communication	99
Annex A (normative) The xDLMS application service element	105
Annex B (normative) Using the COSEM Application Layer in various communication profiles	107
Annex C (informative) AARQ and AARE encoding examples	126
Annex D (informative) Data model and protocol	138
Bibliography	139
INDEX	142

Figure 1 – Client/server relationship in COSEM	10
Figure 2 – Exchanging messages via the communication protocol	11
Figure 3 – The COSEM application layer on the top of various lower layer stacks	11
Figure 4 – A complete communication session in the CO environment	12
Figure 5 – The structure of the COSEM application layers	13
Figure 6 – Structure of the COSEM AL when the server is using SN references	15
Figure 7 – Summary of COSEM application layer services	16
Figure 8 – Normal service sequence for the COSEM-OPEN service	17
Figure 9 – Client side services for application association establishment	19
Figure 10 – Client side services for releasing an application association	23
Figure 11 – Client side data communication services	26
Figure 12 – Client side services for event notification	35
Figure 13 – Server side services for application association establishment	38
Figure 14 – Server side services for releasing an application association	40
Figure 15 – Server side data communications services using LN referencing	44
Figure 16 – Partial state machine for the client side control function	60
Figure 17 – Partial state machine for the server side control function	61
Figure 18 – MSC for successful application association establishment preceded by a successful lower layer connection establishment	63
Figure 19 – Graceful association release using the A-RELEASE service	69
Figure 20 – Graceful release of an application association by disconnection the supporting layer	70
Figure 21 – Aborting an application association following a PH-ABORT indication	71
Figure 22 – MSC for a confirmed GET service in case of success	75
Figure 23 – MSC for a confirmed SET service in case of success	75
Figure 24 – MSC for the SET service in case of failure	76
Figure 25 – MSC for the ACTION service (simplest case)	76
Figure 26 – Long data with the GET service in three data blocks	81
Figure 27 – Long data transfer in three data blocks with the SET service	83
Figure 28 – Long data transfer with the ACTION service	85
Figure 29 – MSC for the ReadRequest/Response services	87
Figure B.1 – Identification/addressing scheme in the 3-layer, connection-oriented, HDLC based communication profile	110
Figure B.2 – Data link layer services provided to and used by the client COSEM application layer	111
Figure B.3 – Data link layer services provided to and used by the server COSEM application layer	112
Figure B.4 – Example: EventNotification triggered by the client	115
Figure B.5 – Multi-drop configuration and its model	116
Figure B.6 – Master/ Slave operation on the multi-drop bus	116
Figure B.7 – COSEM as a standard Internet application protocol	118
Figure B.8 – Examples for lower-layer protocols in the TCP-UDP/IP based profiles	119
Figure B.9 – Identification/addressing scheme in the TCP-UDP/IP based profile(s)	120
Figure B.10 – Summary of TCP/UDP layer services on the client and server side	121
Figure D.1 – The three-step approach of COSEM	138

Table 1 – Mapping between client side LN and server side SN referencing services	37
Table 2 – Application layer services – summary.....	55
Table 3 – Summary of the service parameters in the COSEM-OPEN service primitives.....	56
Table 4 – Summary of the service parameters in the COSEM-RELEASE service primitives	57
Table 5 – Summary of the service parameters in the COSEM-ABORT service primitives	57
Table 6 – Summary of the service parameters in the COSEM GET service primitives	57
Table 7 – Summary of the service parameters in the COSEM SET service primitives.....	58
Table 8 – Summary of the service parameters in the COSEM ACTION service primitives	58
Table 9 – Summary of the service parameters in the COSEM EventNotification service primitives	59
Table 10 – Mapping between the EventNotification and InformationReport services.....	88
Table B.1 – Application associations and data exchange in the 3-layer, connection-oriented, HDLC based profile	113
Table B.2 –Application associations and data exchange in the TCP-UDP/IP based profile	123

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ELECTRICITY METERING – DATA EXCHANGE FOR METER READING, TARIFF AND LOAD CONTROL –

Part 53: COSEM application layer

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a maintenance service concerning the stack of protocols on which the present standard IEC 62056-53 is based.

The IEC takes no position concerning the evidence, validity and scope of this maintenance service.

The provider of the maintenance service has assured the IEC that he is willing to provide services under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the provider of the maintenance service is registered with the IEC. Information may be obtained from:

DLMS¹ User Association

Geneva / Switzerland

www.dlms.ch

International Standard IEC 62056-53 has been prepared by IEC technical committee 13: Equipment for electrical energy measurement and load control.

¹ Device Language Message Specification

This standard cancels and replaces the first edition which was published in 2002. It constitutes a technical revision. The main changes with respect to the previous edition are as follows:

- the protocol of the COSEM-RELEASE service has been changed: depending on the communication profile used, these services may rely on the ACSE A_RELEASE services;
- the parsing order of the AARQ APDU has been changed;
- handling of repeated application association requests has been simplified;
- the Service_Class parameter of the COSEM-OPEN service is now linked to the response-allowed field of the xDLMS-Initiate.request APDU;
- the Service_Class parameter of COSEM services for data exchange using LN referencing is now linked to bit 6 of the Invoke-Id-And-Priority parameter;
- a new, optional EXCEPTION APDU has been introduced. The server may send back this APDU after an erroneous service request;
- a general part about using the COSEM application layer in various communication profiles has been added;
- the description of using the COSEM Application layer in the 3-layer, connection-oriented, HDLC based communication profile has been amended;
- a new, TCP-UDP/IP based communication profile has been defined.

The text of this standard is based on the following documents:

FDIS	Report on voting
13/1387/FDIS	13/1398/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of IEC 62056 series, published under the general title *Electricity metering – Data exchange for meter reading, tariff and load control*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of the publication may be issued at a later date.

ELECTRICITY METERING – DATA EXCHANGE FOR METER READING, TARIFF AND LOAD CONTROL –

Part 53: COSEM application layer

1 Scope

This part of IEC 62056 specifies the COSEM application layer in terms of structure, services and protocols for COSEM clients and servers, and defines how to use the COSEM application layer in various communication profiles.

It defines services for establishing and releasing application associations, and data communication services for accessing the methods and attributes of COSEM interface objects, defined in IEC 62056-62, using either logical name (LN) or short name (SN) referencing.

Annex A describes the xDLMS application service element.

Annex B defines how to use the COSEM application layer in various communication profiles.

Annex C includes encoding examples for APDUs.

Annex D gives an explanation of the role of data models and protocols in electricity meter data exchange.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60050-300:2001, *International Electrotechnical Vocabulary (IEV) – Electrical and electronic measurements and measuring instruments – Part 311: General terms relating to measurements – Part 312: General terms relating to electrical measurements – Part 313: Types of electrical measuring instruments – Part 314: Specific terms according to the type of instrument*

IEC 61334-4-41:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 41: Application protocols – Distribution line message specification*

IEC 61334-6:2000, *Distribution automation using distribution line carrier systems – Part 6: A-XDR encoding rule*

IEC 62051:1999, *Electricity metering – Glossary of terms*

IEC 62051-1:2004, *Electricity metering – Data exchange for meter reading, tariff and load control – Glossary of Terms – Part 1: Terms related to data exchange with metering equipment using DLMS/COSEM*

IEC 62056-21:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange*

IEC 62056-42:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 42: Physical layer services and procedures for connection-oriented asynchronous data exchange*

IEC 62056-46:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol*
Amendment 1²

IEC 62056-47, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 47: COSEM transport layer for IP networks*

IEC 62056-61, Ed.2, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: Object identification system (OBIS)*

IEC 62056-62, Ed.2, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface classes*

ISO/IEC 8649:1996, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element*

ISO/IEC 8650-1:1996, *Information technology – Open systems interconnection – Connection-oriented protocol for the Association Control Service Element: Protocol specification*

ISO/IEC 8824, *Information technology – Abstract Syntax Notation One (ASN.1)*

ISO/IEC 8825, *Information technology – ASN.1 encoding rules*

ISO/IEC 13239:2002, *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*

STD0005 – *Internet Protocol*

Author: J. Postel

Date: September 1981

Also: RFC0791, RFC0792, RFC0919, RFC0922, RFC0950, RFC1112

STD0006 – *User Datagram Protocol*

Author: J. Postel

Date: 28 August 1980

Also: RFC0768

STD0007 – *Transmission Control Protocol*

Author: J. Postel

Date: September 1981

Also: RFC0793

See also Bibliography for other related Internet RFCs.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this part of IEC 62056, the definitions in IEC 60050-300, IEC 62051 and IEC 62051-1 apply.

3.2 Abbreviations

.cnf	.confirm service primitive
.ind	.indication service primitive

² To be published.

.req	.request service primitive
.res	.response service primitive
AA	Application Association
AARE	Application Association REsponse
AARQ	Application Association ReQuest
ACSE	Application Control Service Element
AE	Application Entity
AP	Application Process
APDU	Application layer Protocol Data Unit
API	Application Programming Interface
ARP	Address Resolution Protocol
ASE	Application Service Element
ASO	Application Service Object
ATM	Asynchronous Transfer Mode
A-XDR	Adapted eXtended Data Representation
BER	Basic Encoding Rules
CF	Control function
CO	Connection Oriented
COSEM	COmpanion Specification for Energy Metering
DLMS	Device Language Message Specification
DSAP	Data link Service Access Point
FDDI	Fibre Distributed Data Interface
FTP	File Transfer Protocol
GMT	Greenwich Mean Time
GSM	Global System for Mobile communications
HDLC	High-level Data Link Control
HLS	High Level Security
HTTP	Hypertext Transfer Protocol
IC	Interface Class
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAN	Local Area Network
LLC	Logical Link Control (sub-layer)
LLS	Low Level Security
LPDU	LLC Protocol Data Unit
LSAP	LLC sub-layer Service Access Point
LSB	Least Significant Bit
MAC	Medium Access Control
MD5	Message Digest Algorithm 5
MIB	Management Information Base
MSB	Most Significant Bit
MSC	Message Sequence Chart
OBIS	OBject Identification System

OSI	Open System Interconnection
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
RARP	Reverse Address Resolution Protocol
RFC	Request For Comment
RLRQ	Release Request
RLRE	Release Response
SAP	Service Access Point
SHA-1	Secure Hash Algorithm 1
SNMP	Simple Network Management Protocol
VAA	Virtual Application Association
xDLMS-ASE	extended DLMS Application Service Element

4 The COSEM communications framework

4.1 Client/server type operation, communication profiles

Communication with electricity metering equipment using the COSEM interface object model is based on the **client/server** paradigm where metering equipment³ plays the server role. In this environment, communication takes place always between a client and a server AP: in other words, the server AP provides remote services to the client AP. These services are provided via exchanging messages (SERVICE.requests/.responses) between the client and the server APs, as shown in Figure 1.

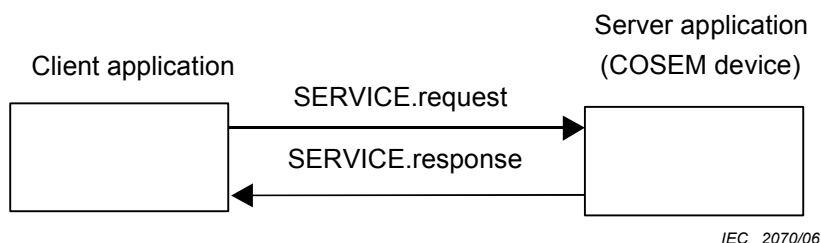


Figure 1 – Client/server relationship in COSEM

In general, the client and the server APs are located in separate devices; exchanging messages is done with the help of the communication protocol as shown in Figure 2.

³ The term metering equipment is an abstraction; consequently the equipment playing the role of a server may be any type of equipment for which this abstraction is suitable.

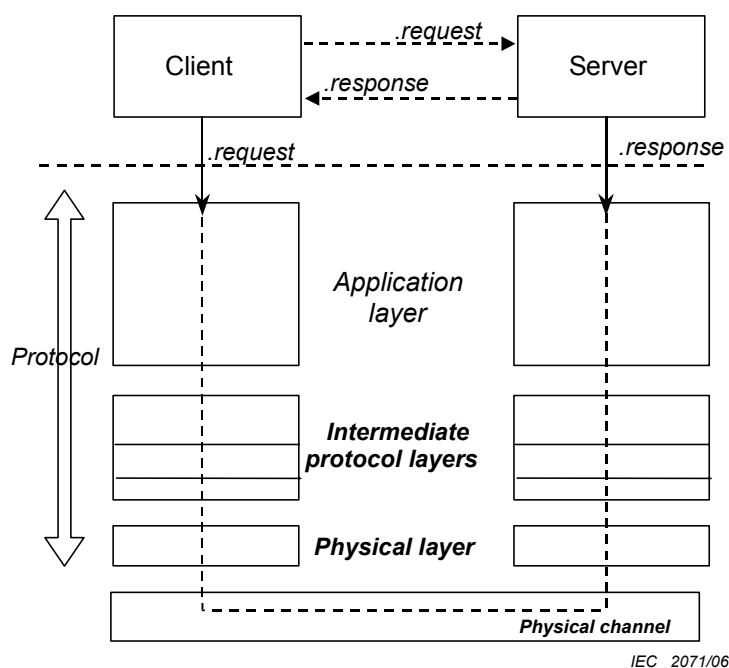


Figure 2 – Exchanging messages via the communication protocol

In general, communication protocols are structured in layers. The client and server COSEM applications use services of the highest protocol layer, that of the application layer: consequently, this is the only protocol layer containing COSEM specific element(s). This is called the xDLMS_ASE. All COSEM interface object related services – the xDLMS application protocol – are provided by this xDLMS_ASE.

Other protocol layers are independent of the COSEM model. Consequently, the COSEM application layer can be placed on the top of a wide variety of lower protocol layer stacks, as shown in Figure 3.

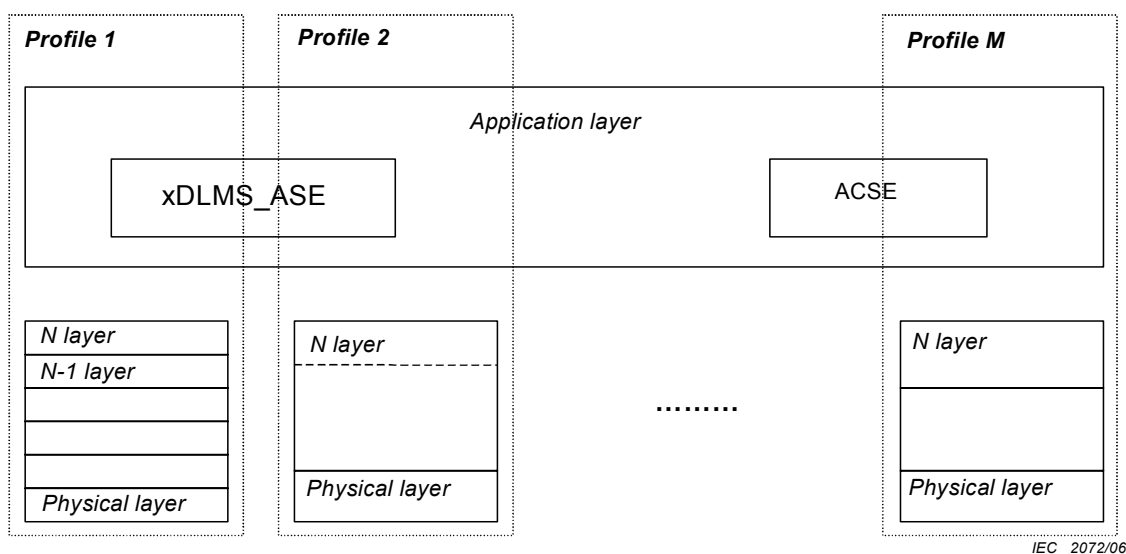


Figure 3 – The COSEM application layer on the top of various lower layer stacks

A complete protocol stack – including the application layer, a physical layer and all protocol layers between these extreme layers – is called a communication profile.

A communication profile is characterized by the protocol layers included, their parameters, and by the type – connection-oriented or connectionless – of the ACSE⁴ included in the application layer.

4.2 Connection (association) oriented operation

The xDLMS application protocol is a connection-oriented protocol. It means that the client and server APs can use the services of the xDLMS_ASE only when these APs are associated⁵. Therefore, in this environment a communication session consists of three phases, as shown in Figure 4.

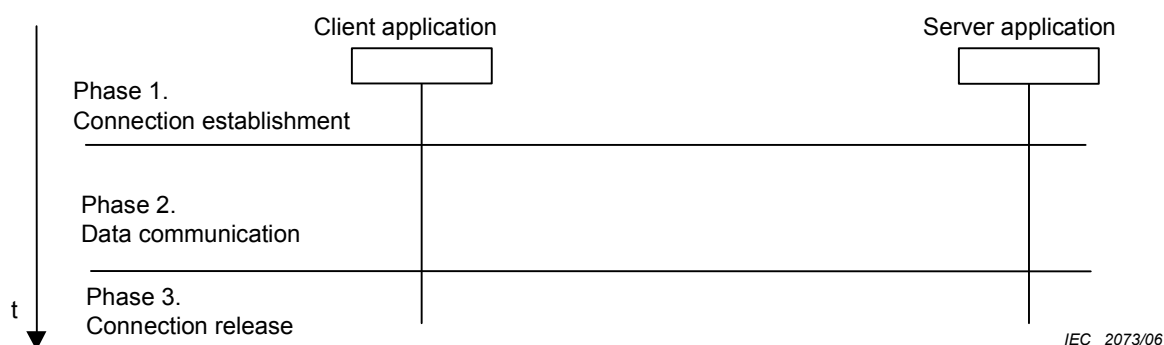


Figure 4 – A complete communication session in the CO environment

In the DLMS/COSEM environment, application association establishment is normally done by using the association request/response services of the standard association control service element. On the other hand, for the purposes of very simple devices, one-way communicating devices and for multicasting and broadcasting, pre-established application associations are also allowed; see 6.3.3. For such associations, there is no need to use the services of the ACSE: a full communication session may include only the data communication phase. (It can be considered that the connection establishment phase has been already done somewhere in the past.)

5 Overview: the COSEM application layer

5.1 Specification method

The COSEM application layer is specified in terms of *structure*, *services*, and *protocols*.

5.2 Application layer structure

The main component of the client and server COSEM application layers is the COSEM ASO, which provides services to the COSEM AP, and uses services provided by the supporting lower layer.

Both the client and server side COSEM ASO contain three mandatory components:

- the ACSE. The task of this element is to establish, maintain, and release application associations. For the purposes of connection-oriented profiles, the connection-oriented ACSE, specified in ISO/IEC 8649 and ISO/IEC 8650-1 is used;

⁴ ACSE = Association Control Service Element

⁵ Application associations can be considered as application level connections.

- the Extended DLMS application service element (xDLMS_ASE). The task of this element is to provide data communication services between COSEM APs. See also Annex A;
- the Control function (CF). This element specifies how the ASO services invoke the appropriate service primitives of the ACSE and the xDLMS ASE and the services of the supporting layer.

NOTE Both the client and the server COSEM ASO may contain other, optional application protocol components.

Figure 5 shows 'minimal' COSEM ASOs, containing only the three mandatory components.

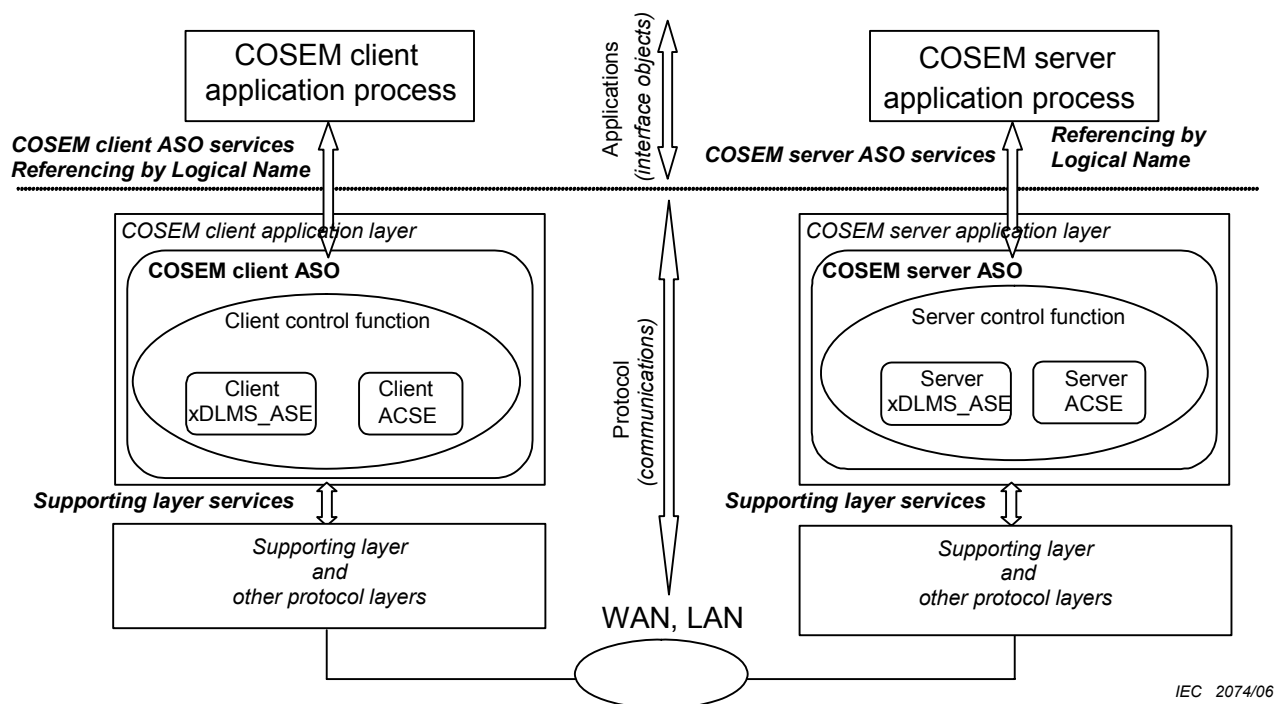


Figure 5 – The structure of the COSEM application layers

The COSEM application layer performs also some functions of the OSI presentation layer:

- for encoding the ACSE APDUS- AARQ, AARE, RLRQ, RLRE – BER encoding is used (ISO/IEC 8825);
- for encoding the APDUs carrying the data communication services, A-XDR encoding is used (IEC 61334-6).

5.3 Service specification

Service specifications cover the services required of, or by the COSEM client and server APs at the logical interfaces with the respective COSEM application layer, using connection-oriented procedures.

Services provided by the COSEM ASO fall into three categories:

- application association establishment and release;
- data communication;
- layer management.

The client and server application layer services are specified in Clause 6.

5.3.1 Services provided for application association establishment and release

These services are the following:

- COSEM-OPEN;
- COSEM-RELEASE;
- COSEM-ABORT.

The COSEM-OPEN service is used during AA establishment phase and relies on the association request/response services of the ACSE. In the case of pre-established application associations (6.3.3) these services are not used.

In certain COSEM communication profiles – for example in the 3-layer, connection-oriented, HDLC based profile – there is a one-to-one relationship between a confirmed AA and the supporting protocol layer connection. In this case, the COSEM-RELEASE service used during the association release phase does not rely on the ACSE A_RELEASE services. Confirmed AAs in these profiles are released simply by disconnecting the corresponding lower layer connection.

Optionally, the COSEM-RELEASE service may rely on the ACSE A_RELEASE service. In some communication profiles, like in the TCP-UDP/IP based profile, using the ACSE A_RELEASE services for releasing COSEM AAs is mandatory.

5.3.2 Data communication services

IEC 62056-62 specifies two referencing methods for COSEM servers: referencing by Logical Names (LN) and referencing by Short Names (SN). Therefore, two distinct service sets are specified for the server side xDLMS_ASE. One set uses exclusively LN references the other set uses exclusively SN references. Thus, these services are the following:

- COSEM interface object attribute-related services: GET, SET for LN referencing and Read, Write, Unconfirmed Write for SN referencing;
- COSEM interface object method-related services: ACTION (LN), Read, Write or UnconfirmedWrite (SN);
- the EventNotification (LN), InformationReport (SN) services.

The services rely on the services of the xDLMS_ASE. Most of these services contain references to attributes or methods of COSEM interface objects.

The service set to be used on the server side during the data communications phase is negotiated during the association establishment phase using the conformance block, see 8.5. It shall not change during the lifetime of the established association. Using LN or SN services within a given AA is exclusive. Therefore, it can be considered that there are two, different server xDLMS_ASE-s: one providing services with Logical name references and another providing services with Short name references. The server application layer shall include one or both of these xDLMS_ASEs.

NOTE A server could use both LN and SN referencing in different AAs.

On the client side, in order to handle the different referencing schemes transparently for the COSEM client AP, the COSEM client application layer provides only one service set, using Logical Name referencing. This has two major consequences:

- using a unique, standardized service set between COSEM client applications and the communications protocol – hiding the particularities of different COSEM servers – allows to specify an Application Programming Interface (API). This is an explicitly specified interface corresponding to this service set for applications running in a given computing environment (e.g. Windows XP, UNIX, etc.) Using this – public – API specification, client applications can be developed without knowledge about particularities of a given server.

- when the COSEM server device does not use LN referencing, the client application layer shall include an additional component. The purpose of this component is to map the LN service set, used by the client AP into/from the service set, used by the server AP. Figure 6 shows the COSEM client application layer when the server is using SN referencing. The additional component is called SN_MAPPER_ASE. See also 6.5.5.2.

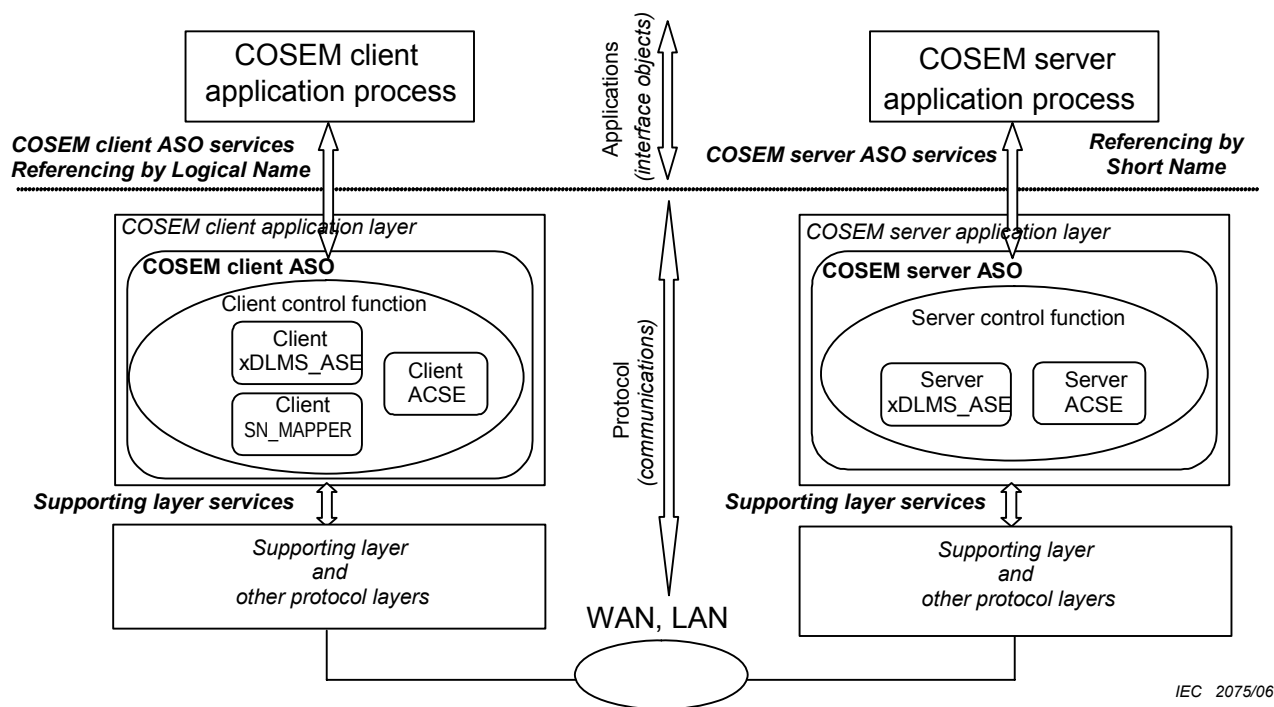


Figure 6 – Structure of the COSEM AL when the server is using SN references

5.4 Layer management services

Layer management services have local importance only. Therefore, specification of these services is not within the scope of this standard. The specific SetMapperTables service is defined in 6.5.5.1.

5.5 Protocol specification

The COSEM application layer protocols specify the procedures for the transfer of information for application association control, authentication (ACSE procedures) and for data exchange between COSEM clients and servers (xDLMS procedures). These procedures are defined in terms of:

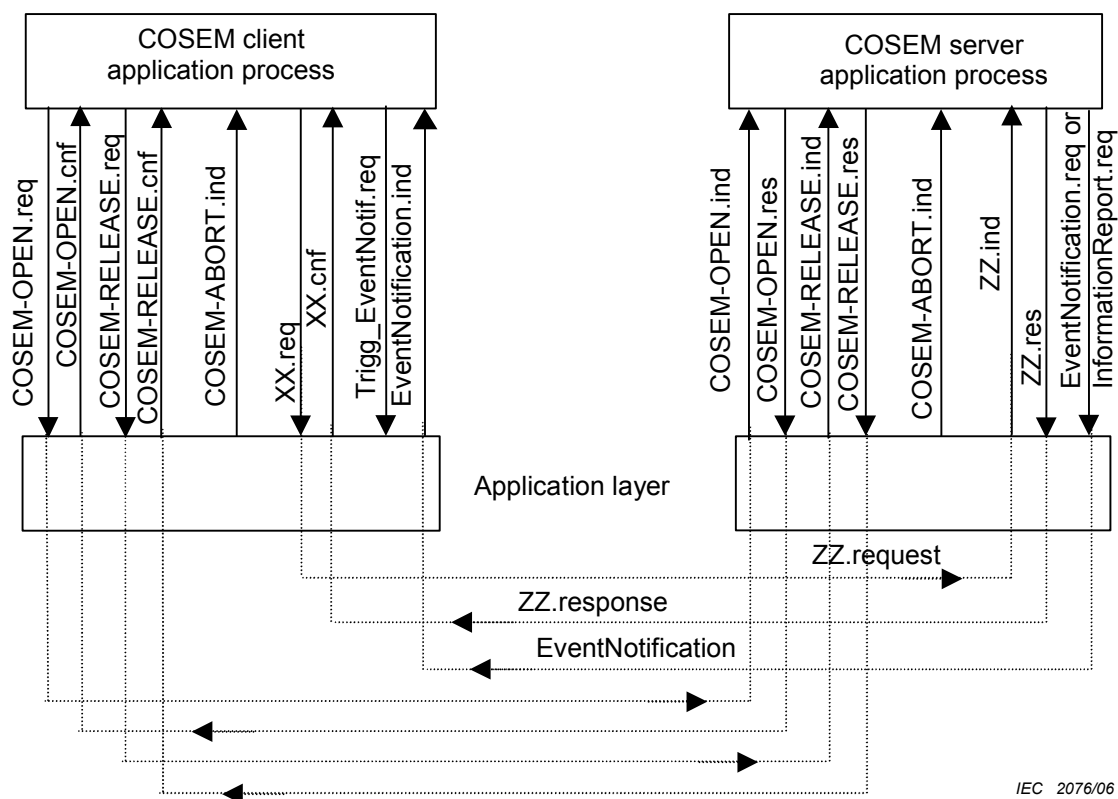
- the interactions between peer ACSE and xDLMS protocol machines through the use of services of the supporting protocol layer;
- the interactions between the ACSE and xDLMS protocol machines and their service user;
- the abstract syntax (ASN.1, ISO/IEC 8824) representation of Application Protocol Data Units (APDUs) is also specified with the application protocols; see Clause 8.

NOTE All COSEM services are operating on an already established physical connection. Establishment of this physical connection is done outside of the COSEM protocol, therefore, it is not within the scope of this standard.

6 COSEM application layer – Service specification

6.1 Summary of services

A summary of the services available at the top of the COSEM application layer is shown in Figure 7.



NOTE XX and ZZ refer to client/server type data communication services. These services may be different on the client side and the server side, if the server does not use LN referencing. See 6.4.

Figure 7 – Summary of COSEM application layer services

6.2 Application association establishment and release

The COSEM-OPEN, COSEM-RELEASE and COSEM-ABORT services are used for the establishment and release of AAs.

The COSEM-OPEN.request service is invoked by the COSEM client AP to establish a confirmed or non-confirmed AA with a COSEM server AP. Invoking this service implies generating a COSEM-OPEN.indication service primitive at the server side⁶. If the association to be established is a confirmed one, the server shall respond to this request by invoking the COSEM-OPEN.response service, which is transferred to the client AP as a remote confirmation (COSEM-OPEN.confirm). This normal opening sequence is shown in Figure 8.

⁶ Before the invocation of the COSEM-OPEN.request service, the physical layers must be connected. Depending on the communication profile, the invocation of the COSEM-OPEN.request service may also imply the connection of the lower layers.

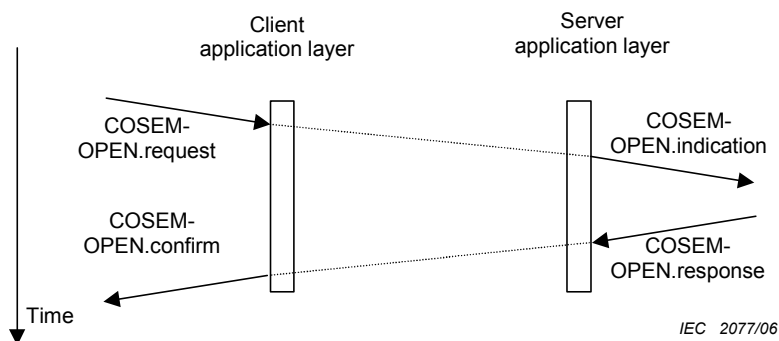


Figure 8 – Normal service sequence for the COSEM-OPEN service

NOTE The COSEM-OPEN.request service may also be locally (negatively) confirmed, for example when the connection of a lower layer is not successful.

The COSEM-RELEASE service is provided for graceful disconnection of an existing application association. As COSEM server application processes are not allowed to request a graceful disconnection, the COSEM-RELEASE.request service is available only for the COSEM client. The nominal service sequence for the COSEM-RELEASE service is the same as shown in Figure 8 for the COSEM-OPEN service, replacing the word 'OPEN' with the word 'RELEASE'.

The ABORT service is used to indicate the disconnection of the physical connection. This service is the same at both sides.

6.3 Special application associations

6.3.1 Confirmed application associations

For the purposes of this standard, the term *confirmed application association* is used to designate an AA, which is established between a client and a server AP with the help of an AARQ/AARE message exchange (see 7.3.1). Establishment of a confirmed AA is always initiated by the client application in invoking the COSEM-OPEN.request service with Service_Class == Confirmed.

After the establishment of a confirmed AA, any xDLMS data communication services using LN referencing can be invoked in a confirmed or an unconfirmed manner, until the association is released.

NOTE xDLMS services using SN referencing are either confirmed (Read, Write) or Unconfirmed (Unconfirmed Write).

6.3.2 Non-confirmed application associations

For the purposes of this standard, the term *non-confirmed application association* is used to designate an AA, which is established without an AARQ/AARE message exchange: only an AARQ shall be sent from the client to the server. Similarly to the confirmed AA, establishment of a non-confirmed AA is also always initiated by the Client application, but in invoking the COSEM-OPEN.request service with Service_Class == Unconfirmed.

After the establishment of a non-confirmed AA, xDLMS data communication services using LN referencing can only be invoked in unconfirmed manner, until the association is released.

NOTE With SN referencing, in non-confirmed AAs only the UnconfirmedWrite service can be used.

As the establishment of such non-confirmed AAs does not require the Server AP to respond to the association request coming from the client, in some cases – for example one-way communications or broadcasting – the establishment of a non-confirmed AA is the only possibility.

6.3.3 Pre-established application associations

Pre-established AAs do not need to be established using the COSEM-OPEN service. It can be considered, that this COSEM-OPEN has already been done (it does not matter how). Consequently, pre-established associations can be considered existing from the moment the lower layers are able to deliver APDUs between the client and the server.

A pre-established AA can be either confirmed or non-confirmed (depending on the way it is pre-established), but in any case it cannot be released. The purpose of this type of association is to simplify data exchange with simple devices (e.g. supporting one-way communication only). The pre-established AA eliminates the need of connection establishment and release (phases 1 and 3 in Figure 4) and only data communication services are used. These must use connectionless services of the supporting lower protocol layers⁷.

6.3.4 Mandatory application associations

The mandatory management logical device in the physical metering device must support an AA with a public client, with the lowest security level.

In any communication profile, the management logical device and the public client must have a reserved identifier/address.

6.4 Data communication

For data communication purposes, the client application layer provides the following set of services (referred to as XX in Figure 7).

- GET service (.request,.confirm);
- SET service (.request,.confirm);
- ACTION service (.request,.confirm).

All these services refer to attributes or methods of COSEM interface objects via logical names.

Received erroneous confirmed service requests are normally simply discarded at the server side. However, in that case, COSEM servers may optionally respond with an EXCEPTION-Response APDU (see 8.6.1), indicating that the previously received service request cannot be correctly processed.

There are also non-client/server type services to support receiving information like alarms from a COSEM server without first requesting it by the client. These are:

- EventNotification service (.indicate);
- Trigger_EventNotification_Sending (.request).

In confirmed AAs, the client application layer obtains knowledge about the referencing method used by the server during the AA establishment phase. In case of a pre-established AA, the client application layer is expected to know the referencing method used by the server before data communication services can be used. When the client AP invokes data communication services, the application layer shall invoke the services and send the APDUs corresponding to the referencing style used by the server (referred to as ZZ in Figure 7).

⁷ Pre-established application associations are not possible in profiles, where the supporting lower protocol layer(s) do not provide connectionless data communication services. As for all application associations, the logical devices must contain an Association LN /SN interface object for the pre-established associations, too.

When the server is also using LN references, the server side service set is the complementary of the client side service set (the same service set, but request services shall be transferred as indication services, and the confirm services are originated as response services).

When the server is using SN references, the service set is as follows:

- READ service (.indication,.response);
- WRITE service (.indication,.response);
- UNCONFIRMED WRITE service (.indication);
- InformationReport service (.request).

As explained in 5.3.2, in order to be able to 'map' between the different service sets, the client application layer shall include an additional protocol component, called 'Client SN_MAPPER'.

The corresponding server application layer shall signal the reception of this (LN or SN referencing) APDU to the server AP. In most cases, the server AP responds to the received request service by invoking the corresponding response service. Upon the reception of the APDU, corresponding to that response invocation, the client application layer shall generate the appropriate logical name referencing service primitive to the client AP.

6.5 Client COSEM application layer services

6.5.1 Application association establishment

6.5.1.1 Overview

Figure 9 shows services provided by the client side application layer for AA establishment. These services are provided by the ACSE.

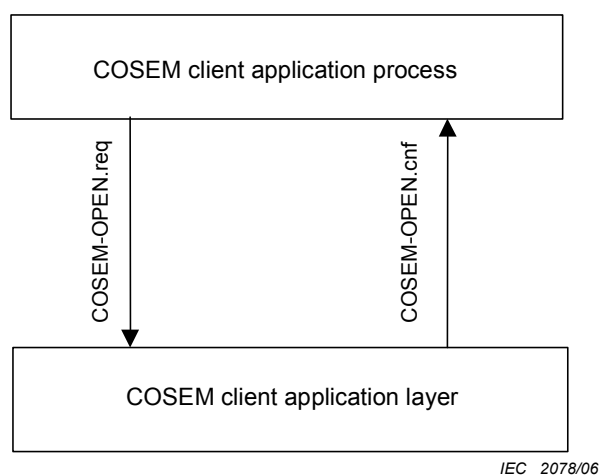


Figure 9 – Client side services for application association establishment

6.5.1.2 COSEM-OPEN.request

Function

This service is invoked by the COSEM client AP to request the establishment of an AA to a remote COSEM server AP.

Service parameters

The semantics of the primitive is as follows:

COSEM-OPEN.request

```
(
    Protocol_Connection_Parameters,
    Dedicated_Key,
    DLMS_Version_Number,
    DLMS_Conformance,
    Client_Max_Receive_PDU_Size,
    ACSE_Protocol_Version,
    Application_Context_Name,
    Application_Ids_and_Titles,
    Security_Mechanism_Name,
    Calling_Authentication_Value,
    Implementation_Information,
    User_Information,
    Service_Class
)
```

The Protocol_Connection_Parameters parameter contains all information necessary to use a lower layer profile, including the communication profile (protocol) identifier and the required addresses. Examples for this parameter are given in Annex B.

The Dedicated_Key, DLMS_Version_Number, DLMS_Conformance and Client_Max_Receive_PDU_Size parameters contain respectively the value of the dedicated-key, the proposed-dlms-version-number, the proposed-conformance and the client-max-receive-pdu-size parameters of the xDLMS-Initiate.request PDU. These parameters are specified in IEC 61334-4-41, and in 8.4 of this standard. Annex C gives some examples of their usage. The xDLMS-Initiate.request PDU shall be inserted in the user-information field of the AARQ APDU to be sent.

The ACSE_Protocol_Version, Application_Context_Name, Application_Ids_and_Titles, Security_Mechanism_Name and the Calling_Authentication_Value parameters shall be inserted into the corresponding fields of the AARQ APDU to be sent.

The xDLMS-ASE and the ACSE provide only the framework for *transporting* this information. To provide and verify that information is the job of the appropriate COSEM AP. Default and allowed values for these fields are defined in 7.3.3.

The Implementation_Information parameter is optional. If present, it shall be inserted into the implementation-information field of the AARQ APDU to be sent.

The User_Information parameter is optional. If present, it shall be passed on to the supporting layer.

The Service_Class parameter indicates whether the service shall be invoked in the confirmed or unconfirmed manner.

Use

The client AP uses this service to initiate the establishment of an AA to a remote server AP.

If the Client AP invokes a COSEM-OPEN.request service with a parameter referring to an already established AA, then the application layer shall locally and negatively confirm this request with the reason that the requested AA already exists. Note, that this is always the case for pre-established AAs.

When the *Protocol_Connection_Parameters* parameter indicates a connection-oriented communication profile (e.g. TCP/IP) but the *Service_class* parameter is set to *Unconfirmed*, the COSEM application layer shall locally and negatively confirm this request, with the reason that the requested AA is not allowed.

When the *Protocol_Connection_Parameters* parameter indicates that one or more supporting lower layer needs to be connected and the requested AA is allowed, the COSEM client shall first establish all required lower layer connections (except for the physical layer connection, which must be already established prior to this service invocation).

When the required supporting lower layer services are available, the COSEM application layer shall construct and send an AARQ APDU to its peer, containing the service parameters received from the AP.

If the COSEM-OPEN.request service is invoked with *Service_class* == Confirmed, the *response-allowed* parameter of the xDLMS-InitiateRequest PDU, inserted in the user-information field of the constructed AARQ shall be set to TRUE. The client application layer is waiting for an AARE response from the server, prior to – positively or negatively – confirming the COSEM-OPEN.request service invocation.

If the COSEM-OPEN.request service is invoked with *Service_Class* == Unconfirmed, the *response-allowed* parameter shall be set to FALSE, the client application layer does not wait any response from the server. In this case, the service invocation shall be locally confirmed.

The protocol for AA establishment is specified in 7.3.1.

6.5.1.3 COSEM-OPEN.confirm

Function

This service is invoked by the COSEM client application layer to indicate whether the previously requested AA is accepted or not.

Service parameters

The semantics of the primitive is as follows:

COSEM-OPEN.confirm

```
(
    Protocol_Connection_Parameters,
    Local_or_Remote,
    Result,
    Failure_type,
    DLMS_Version_Number,
    DLMS_Conformance,
    Server_Max_Receive_PDU_Size,
    ACSE_Protocol_Version,
    Application_Context_Name,
    Application_Ids_and_Titles,
    Security_Mechanism_Name,
    Responding_Authentication_Value,
    Implementation_Information
)
```

The *Protocol_Connection_Parameters* parameter contains all the information required to identify the protocol connection having been established. These parameters identify the participants of the AA requested by the preceding COSEM-OPEN.request service.

The `Local_or_Remote` parameter indicates the origin of the COSEM-OPEN.confirm service primitive invocation. When this parameter is set to `Remote`, the service invocation has been originated by the reception of an AARE APDU from the remote server. Otherwise, the service is locally originated.

In case of a remote confirmation, the `Result` parameter indicates whether the COSEM server AP accepted the requested association or not. In case of local confirmation, the `Result` parameter indicates whether the client side protocol stack accepted the request or not. In the case of non-acceptance (remote or local), the `Failure_type` parameter indicates the reason for not accepting the proposed association.

The `DLMS_Version_Number`, `DLMS_Conformance` and `Server_Max_Receive_PDU_Size` parameters contain respectively the value of the negotiated-dlms-version-number, negotiated-conformance and server-max-receive-PDU-size parameters of the xDLMS-Initiate.response PDU. These parameters are specified in IEC 61334-4-41, and in 8.4 of this standard. Annex C gives some examples for their usage. The xDLMS-Initiate.response PDU is transported in the user-information field of the received AARE APDU.

The `ACSE_Protocol_Version`, `Application_Context_Name`, `Application-Ids_and_Titles`, `Security_Mechanism_Name` and the `Responding_Authentication_Value` parameters carry the value of the corresponding fields of the received AARE APDU.

The `Implementation_Information` parameter, if present, carries the value of the implementation-information field of the received AARE APDU.

Use

The COSEM client application layer uses this service primitive to indicate to the client AP whether the previously proposed AA is accepted or not. It may be generated as a result of a received AARE APDU (remote confirmation). It may also be generated locally in the following cases:

- when the requested AA already exists (this case includes pre-established AAs);
- when the corresponding COSEM-OPEN.request has been invoked with `Service_class == Unconfirmed`;
- when the requested AA is not allowed;
- due to a locally detected error (missing or not correct parameters, failure during the establishment of the requested lower layer connections, missing physical connection, etc).

6.5.2 Application association release

6.5.2.1 Overview

Figure 10 shows the services provided by the client application layer for releasing an existing AA.

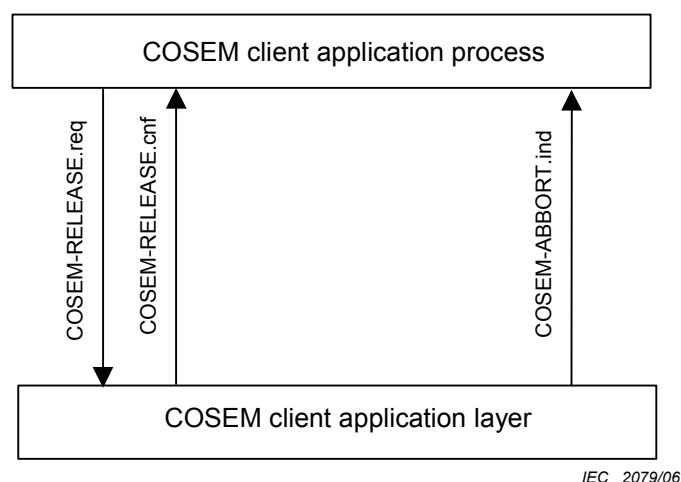


Figure 10 – Client side services for releasing an application association

In certain COSEM communication profiles – for example in the 3-layer, connection-oriented, HDLC based profile – there is a one-to-one relationship between a confirmed AA and the supporting protocol layer connection. In this case, the COSEM-RELEASE services used during the association release phase do not rely on the ACSE A_RELEASE services. Confirmed AAs in these profiles are released simply by disconnecting the corresponding lower layer connection.

This is the mandatory way to release such AAs: in order to request that, the Client application shall invoke the COSEM-RELEASE.request service with no Use_RLRQ_RE parameter or with Use_RLRQ_RE == FALSE.

Note, that as this request shall imply the disconnection of the lower layer connection, there is no mandatory APDU associated to the COSEM-RELEASE service.

The client AP is informed about the result of the requested disconnection via the COSEM-RELEASE.confirm service primitive.

Optionally, the COSEM client AP may invoke the COSEM-RELEASE.request service with Use_RLRQ_RE == TRUE. In this case, the client application layer, instead of disconnecting the lower layer connection (if there is one), shall initiate the release of the referenced AA by sending an RLRQ APDU to the server which may respond to that with an RLRE APDU. The client AP is informed about the result of the association release via the COSEM-RELEASE.confirm service primitive, which, in this case, could be a remotely confirmed service.

Supporting the RLRQ/RLRE APDUs is optional in HDLC based profile.

Any existing AA – except the pre-established ones on the server side – shall be aborted, when the physical connection is intentionally disconnected or if any of the supporting layer connection is disconnected or breaks. A local COSEM-ABORT.indication primitive is provided to inform the AP about this. As physical connection/ disconnection is done outside of the protocol, requesting a physical disconnection is not within the scope of this standard.

6.5.2.2 COSEM-RELEASE.request

Function

This service primitive is invoked by the COSEM client AP to request the release of an existing AA with a remote COSEM server AP.

Service parameters

The semantics of this service primitive is as follows:

COSEM-RELEASE.request

```
(
    Use_RLRQ_RE,
    User_Information
)
```

The Use_RLRQ_RE parameter is optional. If it is not present, it shall be considered as if its value would be equal to FALSE.

The User_Information parameter is optional. If present, it shall be passed on to the supporting layer. Specification of the content of this parameter is not within the scope of this standard.

Use

The Client AP uses this service primitive to gracefully release an existing AA.

When this service is invoked with no Use_RLRQ_RE parameter or with Use_RLRQ_RE == FALSE, the invocation of this service shall not imply sending an APDU.

In communication profiles where the RLRQ service is not supported or only optionally supported, upon the reception of this service invocation the client application layer shall:

- either initiate the disconnection of the corresponding lower layer connection (if the AA to be released is a confirmed AA) by invoking the corresponding XX-DISCONNECT.request service of the supporting lower protocol layer;
- or shall simply release the requested non-confirmed AA and locally confirm the request.

Supporting the COSEM-RELEASE.request service with no Use_RLRQ_RE or with Use_RLRQ_RE == FALSE is mandatory for all COSEM Clients. However, in communication profiles where the RLRQ service is mandatory, invoking the COSEM-RELEASE.request service with no Use_RLRQ_RE or with Use_RLRQ_RE == FALSE may lead to an error⁸: in that case it shall be locally and negatively confirmed.

When this service primitive is invoked with Use_RLRQ_RE == TRUE, the client application layer shall construct an A-RELEASE.request (RLRQ) APDU, and shall send it to the peer server application layer, using the supporting lower layer services. If the RLRQ is sent within a non-confirmed AA, after sending the RLRQ the client application layer shall locally and positively confirm the current request. Otherwise, after sending the RLRQ APDU, the client application layer shall start a time-out, waiting for either the A-RELEASE.response (RLRE) APDU from the server, or for the time-out to be elapsed. On any of these two events, the client application layer shall confirm the COSEM-RELEASE.request primitive.

Supporting the COSEM-RELEASE.request service with Use_RLRQ_RE == TRUE is optional in HDLC based profile.

The protocol for releasing an AA is described in 7.3.6.

6.5.2.3 COSEM-RELEASE.confirm

Function

The COSEM client application layer invokes this service primitive to indicate to the AP whether the previously received request for releasing the AA is accepted.

⁸ When the lower layer connections are not managed by the COSEM application layer.

NOTE The server cannot refuse a release request.

Service parameters

The semantics of the primitive is as follows:

COSEM-RELEASE.confirm

```
(  
    Result,  
    Failure_type,  
    User_Information  
)
```

The Result parameter is the report of the corresponding COSEM-RELEASE.request service. As servers cannot refuse requests of neither releasing AAs nor disconnecting lower layer connections, the value of the Result parameter should normally be SUCCESS.

However, the value of this parameter may also be ERROR: in that case, the Failure_type parameter indicates the reason for that.

The User_Information field may be present only when the service is remotely confirmed. In this case, it contains user specific information carried by the supporting lower protocol layer(s), if this is possible. Specification of its content is not within the scope of this standard.

Use

The COSEM client application layer uses this service primitive to indicate to the client AP the result of the previously requested release of an AA. This service primitive is originated:

- as a result of the invocation of a XX-DISCONNECT.confirm service (where XX is the supporting lower protocol layer); or
- by a locally detected error – missing or not correct parameters, or communication failure at lower protocol layer level; or
- after sending out an RLRQ APDU within a non-confirmed AA (OPTIONAL); or
- by the time-out on waiting for an RLRE APDU from the remote Server (OPTIONAL); or
- as a result of a RLRE APDU received from the remote Server (OPTIONAL).

6.5.2.4 COSEM-ABORT.indication

Function

This service is invoked by the client application layer to indicate to the client AP an unsolicited disconnection of any supporting lower layer connection, including the disconnection of the physical layer.

Service parameters

The semantics of the primitive is as follows:

COSEM-ABORT.indication

```
(  
    Diagnostics  
)
```

The optional Diagnostics parameter shall indicate the possible reason for the disconnection, and may carry lower protocol layer dependent information as well. Specification of the contents of this parameter is not within the scope of this standard.

Use

The client application layer uses this service primitive to indicate to the COSEM client AP that a lower layer connection abort occurred in a non-solicited manner (e.g. the physical line is broken).

NOTE In communication profiles, where the supporting layer connection is not managed by the COSEM application layer, a disconnection of the supporting layer by the manager process leads also to a COSEM-ABORT.indication.

6.5.3 Client/server type data communication services

6.5.3.1 Service overview

Figure 11 shows services provided by the client side application layer during the data communications phase.

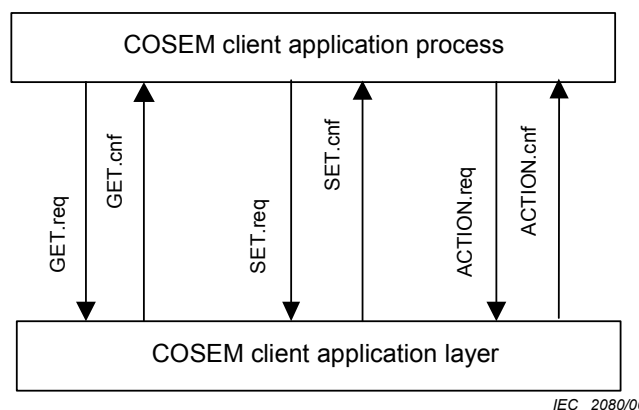


Figure 11 – Client side data communication services

Data communication services rely on the services of the xDLMS_ASE. These services contain references to attributes or methods of COSEM interface objects.

For COSEM servers, two types of referencing are specified in IEC 62056-62: Logical Name (LN) and Short Name (SN). The COSEM client application layer provides only one service set, using logical name referencing. Consequently, when the COSEM server device does not use logical name referencing, the client application layer shall include an additional application protocol component; see in Figure 6. The purpose of this is to 'map' the LN service set into/from the service set used by the server AP.

The service set provided at the COSEM client side is:

- COSEM interface object attribute related services: GET, SET (.request,.confirm);
- COSEM interface object method related service: ACTION (.request,.confirm).

The.request primitive of these services is invoked by the COSEM client AP. The role of the protocol with regard to these services is to transport them as.indication to the COSEM server AP.

NOTE Consequently, a.request APDU is identical to an.indication APDU and a.response APDU is identical to a.confirm APDU. For APDU definitions, see 8.6.

All data communication services within a confirmed AA can be invoked in a confirmed or non-confirmed manner. In case of non-confirmed AAs, data communication services may only be invoked in a non-confirmed manner.

In case of confirmed service invocation, the server AP shall return the confirmation by invoking the corresponding response service primitive. The receipt of this response is indicated to the client AP via the confirm service primitive.

If a confirmed service request cannot be processed by the server (e.g. the request has been received without establishing an AA first, or the request is otherwise erroneous) normally it is discarded by the application layer. However, the COSEM server application layer may optionally send an EXCEPTION-Response APDU to indicate to the client application layer that the service request received could not be processed and the reasons for this.

The optional EXCEPTION-Response APDU is defined in 8.6.1.

Unconfirmed service invocation will not imply response/confirm primitive invocation. The reason for this is to avoid collisions due to potential multiple responses in the case of multicasting and/or broadcasting.

The protocol for confirmed service invocations is described in 7.4.1.1 and for unconfirmed service invocations in 7.4.1.2.

6.5.3.2 GET.request

Function

This service is invoked by client AP to request the value(s) of one or all attributes of one or more COSEM interface object(s) from the remote server AP.

Service parameters

The semantics of the primitive is as follows:

GET.request

```
(
    Invoke_Id,
    Priority,
    Service_Class,
    Request_Type,
    COSEM_Attribute_Descriptor, { COSEM_Attribute_Descriptor, },
    Block_Number
)
COSEM_Attribute_Descriptor
(
    COSEM_Class_Id,
    COSEM_Object_Instance_Id,
    COSEM_Object_Attribute_Id,
    Access_Selection_Parameters
)
```

The Invoke_Id parameter identifies the instance of this service invocation.

The value of the Priority parameter indicates the priority level associated to the received request. There are two priority levels: normal (FALSE) and high (TRUE).

The Service_Class parameter indicates whether the service is invoked in confirmed or unconfirmed manner.

The Request_type parameter indicates the type of the current GET.request service invocation: NORMAL, NEXT or WITH-LIST. A GET.request always starts with a GET.request type NORMAL or WITH-LIST. A GET.request with NEXT type is issued only when the requested data is too long for being transferred in one response APDU. The protocol for non-transparent long data transfer with the GET service is described in 7.4.1.8.2

A GET.request service shall contain one or more COSEM_Attribute_Descriptor parameters, each of them referencing one or all attributes of a COSEM interface object. The COSEM_Attribute_Descriptor parameter is a composite parameter, consisting of the following components:

- the { COSEM_Class_Id, COSEM_Object_Instance_Id } doublet non-ambiguously identifies one and only one COSEM interface object instance;
- the COSEM_Object_Attribute_Id component identifies the attribute(s) of the object instance. COSEM_Object_Attribute_Id = 0 references all attributes of the designated object instance;
- the optional Access_Selection_Parameters component, in case of selective access (see 7.4.1.6) carries the additional data required for the selective GET operation. This parameter can be used only when COSEM_Object_Attribute_Id != 0.

One GET.request invocation may contain as many COSEM_Attribute_Descriptors as the server-max-receive-pdu-size allows. The COSEM_Attribute_Descriptor(s) shall be present only with Request_type == NORMAL or WITH-LIST.

The optional Block_Number parameter is present only when Request_type == NEXT. It carries the number of the last correctly received block of long data.

Use

The client AP uses this service primitive to request the value(s) of one or all attributes of one or more COSEM interface object(s) from the server AP.

6.5.3.3 GET.confirm

Function

This service is invoked by the client application layer to indicate the reception of a Get-response-XX APDU from the COSEM server AP.

Service parameters

The semantics of the primitive is as follows:

```

GET.confirm
(
    Invoke_Id,
    Priority,
    Response_type,
    Result, { Result, }
    Block_Number
)
    
```

The Invoke_Id parameter identifies the instance of this service invocation. Its value shall be equal to the Invoke_Id of the corresponding GET.request service invocation.

The value of the Priority parameter indicates the priority level associated to the response received. The value of this parameter shall be equal to the value of the Priority parameter of the corresponding GET.request service invocation.

The Response_type parameter indicates whether this.confirm service invocation contains the complete response to the previous GET.request service invocation, or it contains only a block of the required data. This parameter shall carry one of the following values:

- NORMAL: the service invocation contains the complete response for a NORMAL GET.request;

- **WITH-LIST:** the service invocation contains the complete response for a GET.request service of type WITH-LIST (including a list of attribute references);
- **ONE-BLOCK:** the service invocation contains one block of the complete response. The Block_Number parameter carries the number of the data block carrying a part of the result as raw data;
- **LAST-BLOCK:** the service invocation contains the last data block of the response.

The Result parameter shall carry either the requested data, or in case of error, the indication of the type of error. If the encoded form of the Result parameter does not fit in one APDU, then it shall be transported in blocks, carried by the result parameter of the Get-Confirm-With-Datablock APDU, of type DataBlock-G. This parameter shall include the block number and the encoded form of the result as raw data or data access result.

The number of Result parameters in the GET.confirm service shall be the same as the number of COSEM_Attribute_Descriptor parameters in the corresponding GET.request service – one response for each request.

Use

The client application layer uses this service primitive to indicate the reception of a Get-Response-XX APDU.

6.5.3.4 SET.request

Function

This service primitive is invoked by the client AP to request the remote server AP to set the value of one or more attributes of a COSEM interface object.

Service parameters

The semantics of the primitive is as follows:

```

SET.request
(
    Invoke_Id,
    Priority,
    Service_Class,
    Request_type,
    COSEM_Attribute_Descriptor, { COSEM_Attribute_Descriptor, },
    Block_Number,
    Data, { Data, }
)

COSEM_Attribute_Descriptor
(
    COSEM_Class_Id,
    COSEM_Object_Instance_Id,
    COSEM_Object_Attribute_Id,
    Access_Selection_Parameters
)

```

The Invoke_Id parameter identifies the instance of this service invocation.

The value of the Priority parameter indicates the priority level associated to the received request. There are two priority levels: normal (FALSE) and high (TRUE).

The Service_Class parameter indicates whether the service is invoked in confirmed or unconfirmed manner.

The Request_type parameter indicates whether the Data parameter of the service primitive carries all the data necessary to set all the attributes referenced by the COSEM_Attribute_Descriptor (list) or only a block of it. It shall be set to one of the following values:

- NORMAL: the service invocation contains the reference to one or all (Attribute_0 feature, see 7.4.1.7.1) attribute(s) of one COSEM interface object and all the required data. The optional Block_Number parameter shall not be present in the service invocation;
- WITH-LIST: the service invocation contains a list of COSEM interface object attribute references and all the required data. The optional Block_Number parameter shall not be present in the service invocation;
- FIRST-BLOCK: the service invocation contains the reference to one or all attribute(s) of one COSEM interface object and the first part of the required data. The Block_Number parameter shall be set to 0001;
- FIRST-BLOCK-WITH-LIST: the service invocation contains a list of COSEM interface object attribute references and the first part of the required data. The Block_Number parameter shall be set to 0001;
- ONE-BLOCK: the service invocation contains only one block of the data. The Block_Number parameter carries the number of the datablock carrying a part of the Data parameter as raw data, and no COSEM_Attribute_Descriptor(s) shall be present;
- LAST-BLOCK: the service invocation contains the last block of the Data. The Block_Number parameter carries the number of this data block, and no COSEM_Attribute_Descriptor(s) shall be present.

NOTE In the case of ONE-BLOCK and LAST-BLOCK Set-Request-With-Datablock APDU is generated.

A SET.request service shall contain one or more COSEM_Attribute_Descriptor parameters, each of them referencing one or more attributes of a COSEM interface object. The COSEM_Attribute_Descriptor parameter is a composite parameter, consisting of the following components:

- the { COSEM_Class_Id, COSEM_Object_Instance_Id } doublet non-ambiguously identifies one and only one COSEM interface object instance;
- the COSEM_Object_Attribute_Id component identifies the attribute(s) of the object instance. COSEM_Object_Attribute_Id = 0 references all attributes of the designated object instance.

The optional Access_Selection_Parameters element, in case of selective access (see 7.4.1.6) carries the additional data required for the selective SET operation. This parameter can be used only when COSEM_Object_Attribute_Id != 0.

One SET.request invocation may contain as many COSEM_Attribute_Descriptors as the server-max-receive-pdu-size allows. The COSEM_Attribute_Descriptor(s) shall be present only when Request_type == NORMAL, Request_type == WITH-LIST or Request_type == FIRST-BLOCK-XXX.

The optional Block_Number parameter is present when Request_type != NORMAL or WITH-LIST. It carries the number of the data block within the current service invocation.

The Data parameter contains the data necessary to set the attributes identified by the Attribute_descriptor parameter(s). If the encoded form of the data does not fit in one APDU, then it shall be transported in blocks, carried by the datablock parameter of the appropriate Set-Request-XX APDU, of type DataBlock-SA. This parameter shall include the block number and the encoded form of the data as raw data. The protocol for long data transfer with the SET service is described in 7.4.1.8.3.

The number of Data parameters in the SET.request service shall be the same as the number of COSEM_Attribute_Descriptors: one Data for each COSEM_Attribute_Descriptor.

Use

The client AP uses this service primitive in order to request the remote server AP to set the value of one or more attributes of one or more COSEM interface objects.

6.5.3.5 SET.confirm*Function*

This service primitive is invoked by the client application layer to indicate the reception of a SET.response from the COSEM server AP.

Service parameters

The semantics of the primitive is as follows:

SET.confirm

```
(
    Invoke_Id,
    Priority,
    Response_type,
    Result { Result, },
    Block_Number
)
```

The Invoke_Id parameter identifies the instance of this service invocation. Its value is equal to the Invoke_Id of the corresponding SET.request service invocation.

The value of the Priority parameter indicates the priority level associated to the received response. The value of this parameter is equal to the value of the Priority parameter of the corresponding SET.request service invocation.

The Response_type parameter indicates whether this.confirm service invocation contains the response for the complete SET.request operation, or it is simply an acknowledge of the previously received data block. This parameter shall carry one of the following values:

- **NORMAL:** the.confirm service contains the confirmation of the previous SET.request operation, which carried a single COSEM interface object attribute reference. The Result parameter carries the result of the required operation;
- **WITH-LIST:** the.confirm service contains the confirmation of the previous SET.request operation, which carried a list of COSEM interface object attribute references. The Result parameter carries the list of results for each required SET operation;
- **ACK-BLOCK:** this value indicates that this.confirm service contains the acknowledgement for the last correctly received data block. The Block_Number parameter carries the number of the received data block;
- **LAST-BLOCK:** the SET.confirm service is invoked with this value after the reception of the last data block of a SET.request service, which carried a reference to a single COSEM interface object attribute. This value indicates that this.confirm service contains the response to the original SET.request service, which has been sent in several blocks. The Result parameter carries the result of the required operation and the Block_Number parameter carries the number of the last data block;
- **LAST-BLOCK-WITH-LIST:** the SET.confirm service is invoked with this value after the reception of the last data block of a SET.request service, which carried a list of COSEM interface object attribute references. This value indicates that this.confirm contains the response to the original SET.request service, which has been sent in several blocks. The Result parameter carries the list of result for each required set operation and the Block_Number parameter carries the number of the last data block.

The number of Result parameters in the SET.confirm service with Response_type == WITH-LIST and LAST-BLOCK-WITH-LIST shall be the same as the number of attribute references in the corresponding SET.request service – one result for each request. Each Result parameter shall carry the result of the corresponding SET.request operation.

Use

The client application layer uses this service primitive to indicate the reception of a Set-Response-XX APDU.

6.5.3.6 ACTION.request

Function

This service is invoked by the client AP to remotely invoke one or more method(s) of one or more COSEM interface object(s) in the remote server AP.

Service parameters

The semantics of the primitive is as follows:

ACTION.request

```
(
    Invoke_Id,
    Priority,
    Service_Class,
    Request_Type,
    COSEM_Method_Descriptor, { COSEM_Method_Descriptor, },
    Block_Number,
    Method_Invocation_Parameters, { Method_Invocation_Parameters, }
)

COSEM_Method_Descriptor
(
    COSEM_Class_Id,
    COSEM_Object_Instance_Id,
    COSEM_Object_Method_Id
)
Method_Invocation_Parameters ::= Data
```

The Invoke_Id parameter identifies the instance of this service invocation.

The value of the Priority parameter indicates the priority level associated to the received request. There are two priority levels: normal (FALSE) and high (TRUE).

The Service_Class parameter indicates whether the service is invoked in a confirmed or an unconfirmed manner.

The Request_type parameter indicates whether the given invocation contains a complete request or only a part of it. It shall be set to one of the following values:

- **NORMAL:** the service invocation contains the reference to one COSEM interface object method and the Method_Invocation_Parameters required for the invocation of this method. The optional Block_Number parameter shall not be present in the service invocation;
- **WITH-LIST:** the service invocation contains a list of COSEM interface object(s) method references and all the required Method_Invocation_Parameters. The optional Block_Number parameter shall not be present in the service invocation;

- **FIRST-BLOCK:** the service invocation contains the reference to one COSEM interface object method and the first part of the required Method_Invocation_Parameters. The Block_Number parameter shall be set to 0001;
- **WITH-LIST-AND-FIRST-BLOCK:** the service invocation contains a list of COSEM interface object methods and the first part of the required Method_Invocation_Parameters. The Block_Number parameter shall be set to 0001;
- **ONE-BLOCK:** the service invocation contains only one block of the Method_Invocation_Parameters. The Block_Number parameter carries the number of the parameter block carrying a part of the Method_Invocation_Parameters parameter, and no COSEM_Method_Descriptor shall be present;
- **LAST-BLOCK:** this value indicates that the current block is the last parameter block to be transferred. The Block_Number parameter carries the number of this parameter block, and no COSEM_Method_Descriptor(s) shall be present;
- **NEXT:** this value indicates that this request contains an acknowledgement for a previously received parameter block, and requests the server to send the next one. The Block_Number parameter carries the number of the last correctly received parameter block.

An ACTION.request service shall contain one or more COSEM_Method_Descriptor parameters, each of them referencing one method of a COSEM interface object. The COSEM_Method_Descriptor parameter is a composite parameter, consisting of the following components:

The {COSEM_Class_Id, COSEM_Object_Instance_Id} doublet non-ambiguously identifies one and only one COSEM interface object instance. The complete COSEM_Method_Descriptor references one method of that object instance: this method is identified by the COSEM_Object_Method_Id component.

The optional Block_Number parameter is present either when the request contains a parameter block to be sent or when the request acknowledges a previously received parameter block (Request_type == NEXT). The Block_Number parameter carries the number of the last received parameter block.

Invoking a method may require additional parameters. The Method_Invocation_Parameters parameter carries the data necessary for the invocation of the method(s) identified by the COSEM_Method_Descriptor parameter. If the encoded form of the Method_Invocation_Parameters does not fit in one APDU, then it shall be transported in blocks, carried by the pblock parameter of the appropriate Action-Request-XX APDU, of type DataBlock-SA. This parameter shall include the block number and the encoded form of the Method_Invocation_Parameters as raw data.

The ACTION.request service shall contain as many Method_Invocation_Parameters then COSEM_Method_Descriptors: one Method_Invocation_Parameters for each COSEM_Method_Descriptor. Therefore, even if the invocation of a method does not require additional parameters, the corresponding Method_Invocation_Parameters component shall be present in the service invocation – but it shall be empty.

The COSEM_Method_Descriptor parameter shall not be present when Request_type == ONE-BLOCK or LAST BLOCK.

Use

This service primitive is used by the client AP to remotely invoke one or more method(s) of one or more COSEM interface object(s) in the remote server AP.

6.5.3.7 ACTION.confirm

Function

This service is invoked by the client application layer to indicate the reception of a ACTION.response from the COSEM server AP.

Service parameters

The semantics of the primitive is as follows:

ACTION.confirm

```
(
    Invoke_Id,
    Priority,
    Response_type,
    Result, { Result, },
    Block_Number,
    Response_Parameters, { Response_Parameters, }
)
```

The Invoke_Id parameter identifies the instance of this service invocation. Its value shall be equal to the Invoke_Id of the corresponding ACTION.request service invocation.

The value of the Priority parameter indicates the priority level associated to the received response. The value of this parameter shall be equal to the value of the Priority parameter of the corresponding ACTION.request service invocation.

The Response_type parameter indicates whether this.confirm service invocation contains the complete response requested by the previous ACTION.request service invocation, it contains only a block of the required data, or it is simply an acknowledge of a previously received block of the ACTION.request service. This parameter shall carry one of the following values:

- NORMAL: the service invocation contains the complete response for a NORMAL ACTION.request which carried a single COSEM interface object method reference;
- WITH-LIST: the service invocation contains the complete response for a WITH-LIST ACTION.request service, including a list of COSEM interface object method references;
- ONE-BLOCK: the service invocation contains only one block of the complete response. The Block_Number parameter carries the number of the parameter block carrying a part of the response as raw data;
- LAST-BLOCK: this value indicates that the service invocation contains the last block of the response as raw data;
- NEXT: this value indicates that service invocation contains an acknowledgement for the previously received parameter block and requests the client to send the next one. The Block_Number parameter carries the number of the last correctly received parameter block.

The Result parameter carries the result of the invocation of the COSEM interface object method(s).

The Response_Parameters carries the optional data to be returned, as a result of the invocation of the COSEM interface object methods.

The number of Result and Response_Parameters parameters in the ACTION.confirm service primitive with Response_type == WITH-LIST or a.confirm service which is sent in several parameter blocks shall be the same as the number of COSEM interface object method references in the corresponding ACTION.request service – one Result and Response_Parameter for each request.

If the encoded form of the Result and Response_Parameters does not fit in one APDU, then it shall be transported in block, carried by the pblock parameter of the Action-Response-With-Pblock APDU, of type DataBlock-SA. This parameter shall include the block number and the encoded form of the Result and Response_parameters as raw data.

Use

The client application layer uses this service primitive to indicate the reception of an Action-Response-XX APDU.

6.5.4 Client side services for event notification

Figure 12 shows services provided by the client side application layer for event notification.

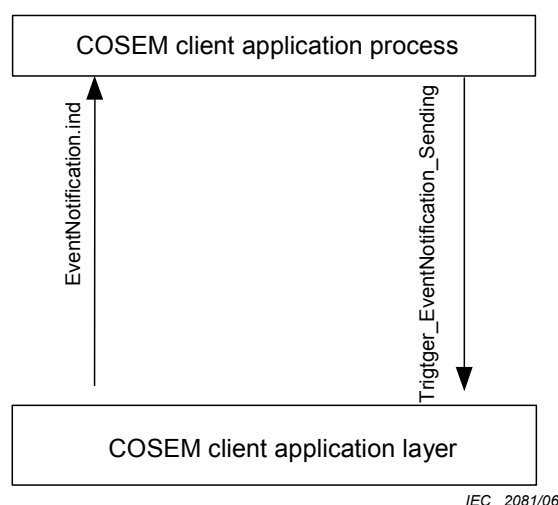


Figure 12 – Client side services for event notification

The EventNotification service is the only non-client/server type service provided in COSEM. Using the EventNotification.request service, the server AP is able to send an unsolicited notification of the occurrence of an event to the remote client AP. Reception of the EventNotification message is indicated to the client AP via the EventNotification.indication primitive. The protocol is described in 7.4.1.3.

In some cases, the supporting lower layer protocol(s) do (does) not allow sending a protocol data unit in a real, unsolicited manner. In these cases, the client shall explicitly solicit sending an EventNotification frame, by invoking the Trigger_EventNotification_sending service primitive.

6.5.4.1 EventNotification.indication

Function

This service is invoked by the client application layer to indicate the reception of an EVENT-NOTIFICATION.indication from the COSEM server SAP.

Service parameters

The semantics of the primitive is as follows:

EventNotification.indication

```

(
    Time,
    Application-Addresses,
    COSEM_Attribute_Descriptor,
    Attribute_Value
)
  
```

```
COSEM_Attribute_Descriptor
(
    COSEM_Class_Id,
    COSEM_Object_Instance_Id,
    COSEM_Object_Attribute_Id
)
```

The optional Time parameter indicates the time assigned to the event by the server.

The Application_Addresses parameter is optional. It is present only when the EventNotification A-PDU is received outside of an established AA. In this case, it contains all protocol specific parameters required to identify the sender and destination APs.

The { COSEM_Class_Id, COSEM_Object_Instance_Id, COSEM_Object_Attribute_Id } triplet identifies non-ambiguously one and only one attribute of a COSEM interface object instance.

The Attribute_Value parameter carries the value of this attribute. More information about the notified event may be obtained by interrogating this COSEM interface object.

Use

The client application layer uses this service primitive to indicate the reception of an EVENT-NOTIFICATION.indication to the AP.

6.5.4.2 Trigger_EventNotification_Sending.request

Function

This service is invoked by the client in order to trigger the server to send the frame carrying the EVENT-NOTIFICATION-Request APDU.

NOTE This service is necessary in case of communication profiles, when the server is not able to send a real non-solicited EventNotification message.

Service parameters

The semantics of the primitive is as follows:

```
Trigger_EventNotification_Sending.req
(
    Protocol_Parameters
)
```

The Protocol_Parameters parameter contains all lower protocol dependent information, which is required for triggering the server to send out an eventually pending frame containing an EVENT-NOTIFICATION-Request APDU. This information includes the protocol identifier, and all the required lower layer parameters.

Use

Upon the reception of a Trigger_EventNotification_Sending.request service invocation from the client AP, the client application layer shall invoke the corresponding supporting layer service to send a trigger message to the server.

6.5.5 Client side layer management services

This subclause defines a special layer management service, used to manage the short name mapper application service element. This client side service is necessary only if the server uses SN referencing. All other layer management services are not within the scope of this standard.

6.5.5.1 SetMapperTable.request

Function

This service is invoked by the client AP to provide mapping information to the Client SN_MAPPER ASE. This service does not cause any data transmission between the client and the server. This service is necessary only if on the server side SN referencing is used.

Service parameters

The semantics of the primitive is as follows:

SetMapperTable.request

```
(
    Mapping_table
)
```

The Mapping_table parameter contains the contents of the attribute “object_list” for the requested server and AA. The structure of the content is defined in IEC 62056-62.

Use

The client AP uses this service primitive, in order to enhance the efficiency of the mapping process if SN referencing is used.

6.5.5.2 Mapping client services for servers using Short names

For servers using SN referencing, the services listed above are mapped to the corresponding xDLMS services (compare IEC 61334-4-41) by the client Control function (see Figure 6) in the following manner:

Table 1 – Mapping between client side LN and server side SN referencing services

Client side xDLMS Service (LN ref.)	Server side xDLMS Service (SN ref.)
GET.request	ReadRequest
GET.confirm	ReadResponse
SET.request (Service_Class="confirmed")	WriteRequest
SET.request (Service_Class="unconfirmed")	UnconfirmedWriteRequest
SET.confirm	WriteResponse
ACTION.request (Service_Class="unconfirmed")	UnconfirmedWriteRequest
ACTION.request (Service_Class="confirmed")	Action with return parameters: ReadRequest VariableAccessSpecification:= parametrised access (IEC 61334-4-41) Selector:= 0; If no method invocation parameters are supplied: Parameter:= null-data Action without return parameters: WriteRequest If no method invocation parameters are supplied: Data:= null-data
ACTION.confirm	ReadResponse If no data is returned then: data:= null-data.
EVENTNOTIFICATION.indication	InformationReportRequest

Details about the mapping of the logical names to short names are given in IEC 62056-62.

6.6 Server COSEM application layer services

6.6.1 Application association establishment

6.6.1.1 Overview

Figure 13 shows the services provided by the server application layer for AA establishment. These services are provided by the ACSE.

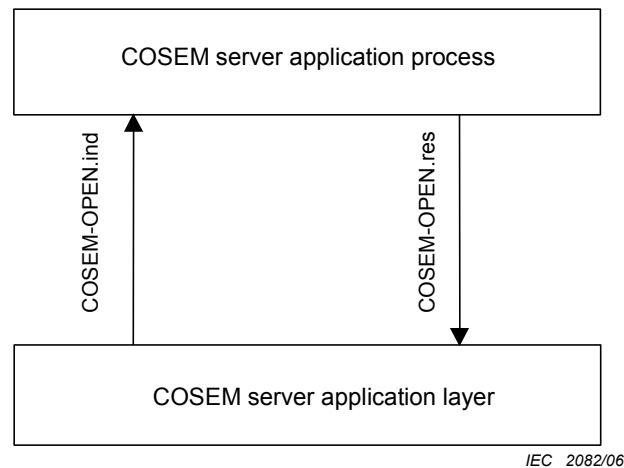


Figure 13 – Server side services for application association establishment

6.6.1.2 COSEM-OPEN.indication

Function

This service is invoked by the server side of the application layer following the receipt of an AARQ APDU, to indicate to the COSEM server AP that the peer (client) AP requested the establishment of an AA.

Service parameters

The semantics of the primitive is as follows:

COSEM-OPEN.indication

```

(
    Protocol_Connection_Parameters,
    Dedicated_Key,
    DLMS_Version_Number,
    DLMS_Conformance,
    Client_Max_Receive_PDU_Size,
    ACSE_Protocol_Version,
    Application_Context_Name,
    Application-Ids_and_Titles,
    Security_Mechanism_Name,
    Calling_Authentication_Value,
    Implementation_Information,
    User_Information,
    Service_Class
)
    
```

The Protocol_Connection_Parameters parameter contains all information necessary to use the supporting layer, including the profile (protocol) identifier and the required addresses. Examples for this parameter are given in Annex B.

The Dedicated_Key, DLMS_Version_Number, DLMS_Conformance and Client_Max_Receive_PDU_Size parameters contain respectively the value of the dedicated-key, the proposed-dlms-version-number, the proposed-conformance and the client-max-receive-pdu-size parameters of the xDLMS-Initiate.request PDU. These parameters are specified in IEC 61334-4-41 and in 8.4 of this standard. Annex C gives some examples for their usage. The xDLMS-Initiate.request PDU shall be inserted in the user-information field of the AARQ APDU received.

The ACSE_Protocol_Version, Application_Context_Name, Application_Ids_and_Titles, Security_Mechanism_Name and the Calling_Authentication_Value parameters are carried by the corresponding fields of the received AARQ APDU.

The xDLMS-ASE and the ACSE provide only the framework for transporting this information. To provide and verify that information is the job of the appropriate COSEM AP. Default and allowed values for these fields are defined in 7.3.7.

The Implementation_Information parameter, if present, carries the value of the implementation-information field of the received AARQ APDU.

The User_Information parameter is optional. When present, it contains the information sent by the Client AP using the same parameter in the corresponding.request primitive.

The Service_Class parameter indicates whether the service is invoked in confirmed or unconfirmed manner.

Use

This service is used by the server side application layer to indicate the reception of a correctly formatted AARQ APDU to the COSEM server AP. In order to be able to receive this APDU, lower layer protocol connections – if it is required – have to be already established.

The protocol for AA establishment is described in 7.3.1.

6.6.1.3 COSEM-OPEN.response

Function

This service is invoked by the server AP to indicate whether the previously proposed AA is accepted or not.

Service parameters

The semantics of the primitive is as follows:

COSEM-OPEN.response

```
(
    Protocol_Connection_Parameters,
    Result,
    Failure_type,
    DLMS_Version_Number,
    DLMS_Conformance,
    Server_Max_Receive_PDU_Size,
    ACSE_Protocol_Version,
    Application_Context_Name,
    Application_Ids_and_Titles,
    Security_Mechanism_Name,
    Responding_Authentication_Value,
    Implementation_Information
)
```

The `Protocol_Connection_Parameters` parameter contains all information necessary to use the supporting layer, including the profile (protocol) identifier and the required addresses. Examples for this parameter are given in Annex B.

The `Result` parameter indicates whether the COSEM server AP accepted the association request or not.

In the case of non-acceptance, the `Failure_type` parameter indicates the reason for not accepting the proposed AA.

The `DLMS_Version_Number`, `DLMS_Conformance` and `Server_Max_Receive_PDU_Size` parameters contain respectively the value of the negotiated-dlms-version-number, negotiated-conformance and server-max-receive-pdu-size parameters of the xDLMS Initiate.response PDU. These parameters are specified in IEC 61334-4-41, and in 8.4 of this standard. Annex C gives some examples for their usage. The xDLMS-Initiate.request PDU shall be inserted in the user-information field of the AARE APDU to be sent.

The `ACSE_Protocol_Version`, `Application_Context_Name`, `Application-Ids_and_Titles`, `Security_Mechanism_Name` and the `Responding_Authentication_Value` parameters shall be inserted into the corresponding fields of the AARE APDU to be sent.

The `Implementation_Information` parameter, if present, shall be inserted in the implementation-information field of the AARE APDU to be sent.

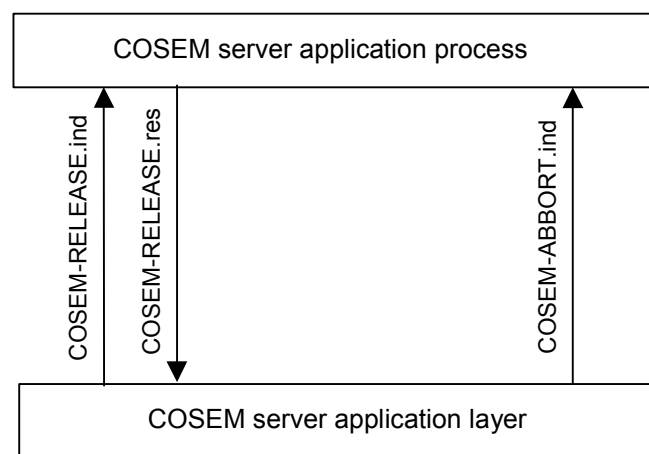
Use

This service primitive is used by the COSEM server AP to indicate to the application layer whether the previously proposed AA is accepted or not.

If the designated AA is confirmed (it was requested by an AARQ with response-allowed == TRUE), the invocation of this service shall imply sending an AARE APDU to the requestor client application layer, otherwise no AARE shall be sent.

6.6.2 Application association release

Figure 14 shows the services provided by the server side application layer for disconnecting an AA.



IEC 2083/06

Figure 14 – Server side services for releasing an application association

In certain COSEM communication profiles – for example in the 3-layer, connection-oriented, HDLC based profile – there is a one-to-one relationship between a confirmed AA and the supporting protocol layer connection. In this case, the COSEM-RELEASE and COSEM-ABORT services used during the association release phase do not rely on the ACSE services. Confirmed AAs in these profiles are released simply by disconnecting the corresponding lower layer connection.

This is the mandatory way to release such AAs.

This way of releasing an AA is requested by the client AP by invoking the COSEM-RELEASE.request service with no Use_RLRQ_RE parameter or with Use_RLRQ_RE == FALSE. In receiving this request, the client application layer shall disconnect the supporting layer connection.

Disconnecting the supporting layer connection shall imply the server application layer to receive an XX-DISCONNECT.indication with REASON == REMOTE from the supporting protocol layer. The reception of this service makes the server application layer to indicate the request for releasing the corresponding AA to the server AP with the help of the COSEM-RELEASE.indication service primitive. The server application shall respond to this with the invocation of the COSEM-RELEASE.response primitive. This response will imply the server application layer to respond to the XX-DISCONNECT.indication and release the designated AA.

Optionally, the COSEM client AP may invoke the COSEM-RELEASE.request service with Use_RLRQ_RE == TRUE. In this case, the client application layer, instead of disconnecting the supporting layer connection (if there is one), shall initiate the release of the referenced AA by sending an RLRQ APDU to the Server.

The reception of this RLRQ APDU has the same effect as the XX-DISCONNECT.indication: the server application layer shall indicate the received release request to the server AP with the help of the COSEM-RELEASE.indication service primitive. The server AP shall respond to this with the COSEM-RELEASE.response.

If the AA to be released is a confirmed AA, the reception of this COSEM-RELEASE.response will imply sending an RLRE APDU to the Client, otherwise no RLRE shall be sent.

Supporting the RLRQ/RLRE APDUs is optional.

Any existing AA – except the pre-established ones on the server side – shall be aborted, when the physical connection is intentionally disconnected or if any of the supporting layer connection is disconnected or breaks. A local COSEM-ABORT.indication primitive is provided to inform the AP about this. As physical connection/disconnection is done outside of the protocol, requesting a physical disconnection is not within the scope of this standard.

6.6.2.1 COSEM-RELEASE.indication

Function

This service primitive is invoked by the COSEM server application layer to indicate to the server AP a supporting layer disconnection indication.

Service parameters

The semantics of the primitive is as follows:

```

COSEM-RELEASE.indication
(
    Use_RLRQ_RE,
    User_Information
)

```

The Use_RLRQ_RE parameter is optional. If it is not present, it is considered, as if the value of this parameter would be equal to FALSE.

The User_Information parameter is optional. When it is present, it shall contain User-specific information carried by the supporting lower protocol layer(s). Specification of the contents of this parameter is not within the scope of this standard.

Use

This service is used by the server application layer to indicate to the server AP that a graceful release of the AA has been requested. The server must accept this request.

When this service is invoked with no Use_RLRQ_RE parameter or with Use_RLRQ_RE == FALSE, the invocation of this service is generated upon the reception of a supporting protocol layer disconnect indication. In this case, the server AP shall respond to this indication with a COSEM-RELEASE.response primitive with no Use_RLRQ_RE or with Use_RLRQ_RE == FALSE, too.

Supporting the COSEM-RELEASE.indication service with no Use_RLRQ_RE or with Use_RLRQ_RE == FALSE is mandatory for all COSEM Servers.

Optionally, COSEM servers may support releasing AAs with the help of the RLRQ/RLRE APDUs. COSEM servers supporting this operation shall invoke the COSEM-RELEASE.indication service with Use_RLRQ_RE == TRUE on the receipt of an RLRQ APDU. In this case, if the COSEM-RELEASE.indication has been received in a confirmed AA, the server application shall respond to this indication with a COSEM-RELEASE.response primitive with Use_RLRQ_RE == TRUE, too.

6.6.2.2 COSEM-RELEASE.response

Function

This service primitive is invoked by the COSEM server AP to indicate to the application layer whether the previously received request for releasing the AA has been accepted.

NOTE The server cannot refuse a received request for disconnection.

Service parameters

The semantics of the primitive is as follows:

COSEM-RELEASE.response

```
(
    Result,
    User_Information,
    Use_RLRQ_RE
)
```

The Result parameter indicates whether the server AP can accept the previous COSEM-RELEASE.request or not. Its value depends on whether the AA, the release of which has been requested, was existing or not.

If the User_Information parameter is present, it shall be passed on to the supporting protocol layer. Specification of its content is not within the scope of this standard.

The Use_RLRQ_RE parameter is optional. If it is not present, it is considered, as if the value of this parameter would be equal to FALSE. If this parameter is present and its value is TRUE, then the primitive is a response to a COSEM-RELEASE.indication, which has been originated by the reception of a RLRQ APDU. Otherwise, the origin of the COSEM-RELEASE.indication was the reception of a disconnect indication with REASON == REMOTE from the supporting protocol layer.

Use

This service primitive is invoked by the server AP to respond to a previously received COSEM-RELEASE.indication. Upon the reception of this invocation, the server application layer shall process the release of the corresponding AA, and perform the appropriate actions as well. These actions can be one of the following:

- invoking the XX-DISCONNECT.response service of the supporting protocol layer, if the Use_RLRQ_RE parameter is not present or its value is FALSE;
- constructing and sending an RLRE APDU to the peer client application layer, if the Use_RLRQ_RE parameter is present and its value is TRUE, and the released AA is a confirmed one. Supporting this operation is optional.

6.6.2.3 COSEM-ABORT.indication*Function*

This service primitive is invoked by the COSEM server application layer to indicate to the server application that the AA is aborted because of the supporting lower protocol connection is shut down. It can be the result either of an unsolicited disconnection of any supporting lower layer connection, including the disconnection of the physical layer, or of the action of another application entity, such as the connection manager application, present in some profiles.

Service parameters

The semantics of the primitive is as follows:

COSEM-ABORT.indication

```
(
    Diagnostics
)
```

The optional Diagnostics parameter shall indicate the possible reason for the physical disconnection, and can carry lower protocol layer dependent information. Specification of the contents of this parameter is not within the scope of this standard.

Use

The server application layer uses this service primitive to indicate to the COSEM server AP that a lower layer connection abort occurred in a non-solicited manner. The origin of the abort can be an external event (e.g. the physical line is broken), or an action of a supporting layer connection manager AP, present in some profiles.

6.6.3 Client/server type data communication services**6.6.3.1 Service overview**

Services provided during the data communications phase rely on services of the xDLMS_ASE. These services contain references to attributes or methods of COSEM interface objects. IEC 62056-62 defines two different types of referencing, by logical name (LN) and by short name (SN). Therefore, two different server xDLMS_ASE-s – thus two different server application layers – are specified. These server side application layers provide two different sets of services. One set of services (GET, SET, ACTION and EventNotification) is using exclusively LN references. The other set of services (Read, Write, Unconfirmed Write, InformationReport) is using exclusively SN references.

However, during the lifetime of an established AA, there is only one server xDLMS_ASE present in the COSEM server application layer. The type of this xDLMS_ASE is negotiated during the connection establishment phase and only the selected xDLMS_ASE is present.