The (COSEM_Class_Id, COSEM_Object_Instance_Id, COSEM_Object_Attribute_Id) triplet references 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.

If the encoded form of the request does not fit in a single APDU, it can be transported in data blocks using the general block transfer mechanism.

Use

A possible logical sequence of the EventNotification service primitives is illustrated in Figure 35 f) and g).

The .request primitive is invoked by the server AP to send the value of a COSEM interface object attribute to the remote client AP. Upon reception of the .request primitive, the Server AL builds the EventNotificationRequest APDU.

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 has to explicitly solicit sending an EventNotification frame, by invoking the Trigger_EventNotification_Sending service primitive.

The EventNotification.indication primitive is generated by the client AL upon reception of an EventNotificationRequest APDU.

The protocol for the EventNotification service is specified in 7.3.8.

6.12 The TriggerEventNotificationSending service

Function

The function of the TriggerEventNotificationSending service is to trigger the server by the client to send the frame carrying the EventNotification.request APDU.

This service is necessary in the case of protocols, when the server is not able to send a real non-solicited EventNotification.request APDU.

Semantics of the service primitives

The TriggerEventNotificationSending.request service primitive shall provide parameters as shown in Table 51.

Table 51 – Service parameters of the TriggerEventNotificationSending.request service primitive

	.request
Protocol_Parameters	M

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

Use

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

6.13 Variable access specification

Variable_Access_Specification is a parameter of the xDLMS Read / Write / UnconfirmedWrite InformationReport .request / .indication service primitives. Its variants are shown in Table 52:

- Variable_Name identifies a DLMS named variable;
- Parameterized_Access provides the capability to transport additional parameters;
- Block_Number_Access transports a block number;
- Read_Data_Block_Access transports block transfer control information and raw data;
- Write_Data_Block_Access transports block transfer control information.

The use of the different variants depends on the service and it is described in the respective SN service specifications.

Table 52 – Variable Access Specification

Variable_Access_Specification	Read .request	Write .request	Unconfirmed Write.request	Information Report
Kind_Of_Access	M	M	M	M
Variable_Name	S	S	S	M
Detailed_Access	Not used in DLMS/COSEM			
Parameterized_Access	S	S	S	–
Variable_Name	M	M	M	
Selector	U	U	U	
Parameter	U	U	U	
Block_Number_Access	S	–	–	–
Block_Number	M			
Read_Data_Block_Access	S	–	–	–
Last_Block	M			
Block_Number	M			
Raw_Data	M			
Write_Data_Block_Access	–	S	–	–
Last_Block		M		
Block_Number		M		

6.14 The Read service

Function

The Read service is used with SN referencing. It is a confirmed service. Its functions are:

- to read the value of one or more COSEM interface object attributes. In this case, the encoded form of the .request service primitive shall fit in a single APDU. The result can be delivered in a single response, or – if it is too long to fit in a single response – in multiple responses, with block transfer;

- to invoke one or more COSEM interface object methods, when return parameters are expected. In this case, if either the .request (including the method references and the method invocation parameters) or the .response service primitive (including the results and return parameters) is too long to fit in a single APDU, then block transfer with multiple requests and/or responses can be used.

The Read service is specified in IEC 61334-4-41:1996, 10.4 and Annex A. For completeness and for consistency with the specification of services using LN referencing, the specification is reproduced here, together with the extensions made for DLMS/COSEM.

Semantics of the service primitives

The Read service primitives shall provide service parameters as shown in Table 53.

Table 53 – Service parameters of the Read service

	.request	.indication	.response	.confirm
Variable_Access_Specification { Variable_Access_Specification }	M	M (=)	–	–
Variable_Name	S	S (=)		
Parameterized_Access	S	S (=)		
Variable_Name	M	M (=)		
Selector	U	U (=)		
Parameter	U	U (=)		
Read_Data_Block_Access	S	S (=)		
Last_Block	M	M (=)		
Block_Number	M	M (=)		
Raw_Data	M	M (=)		
Block_Number_Access	S	S (=)		
Block_Number	M	M (=)		
Result (+)			S	S (=)
Read_Result { Read_Result }	–	–	M	M (=)
Data			S	S (=)
Data_Access_Error			S	S (=)
Data_Block_Result			S	S (=)
Last_Block			M	M (=)
Block_Number			M	M (=)
Raw_Data			M	M (=)
Block_Number			S	S (=)
Result (–)	–	–	S	S (=)
Error_Type			M	M (=)
NOTE For security parameters, see Table 40.				

The use of the different variants of the Variable-Access-Specification service parameter of the Read.request service primitive and the different choices of the Read.response primitive are shown in Table 54.

If the encoded form of the response does not fit in a single APDU, it can be transported in data blocks using the general block transfer mechanism.

Table 54 – Use of the Variable_Access_Specification variants and the Read.response choices

Read.request Variable_Access_Specification		Read.response CHOICE	
Variable_Name (Variable_Name}	References a list[1] of COSEM object attributes.	Data {Data}	Delivers the value of the attribute(s) referenced.
		Data_Access_Error {Data_Access_Error}	Provides the reason for the read to fail.
		Data_Block_Result	Delivers block transfer control information and one block of raw data.
Parameterized_Access {Parameterized_Access}	References a list[1] of COSEM object attributes to be read selectively.	Data {Data}	As above.
		Data_Access_Error {Data_Access_Error}	
		Data_Block_Result	
	References a list[1] of COSEM object methods, with method invocation parameters.	Data {Data}	Delivers the method invocation return parameters. NOTE If parameters are returned, this implies that the method invocation succeeded.
		Data_Access_Error {Data_Access_Error}	Provides the reason for the method invocation to fail.
		Data_Block_Result	As above.
Read_Data_Block_ Access	Carries block transfer control information and one part of encoded form of the COSEM method references and method invocation parameters.	Block_Number	Carries the number of the latest data block received.
Block_Number_Access	Carries the number of the latest data block received.	Data_Block_Result	As above.

NOTE The same Read.response choice can be present more than once, to show the possible responses to each request.

[1] A list may have one or more elements.

The Read.request service primitive may have one or more Variable_Access_Specification parameters.

- the Variable_Name variant is used to reference a complete COSEM object attribute to be read. The request may include one or more variable names;
- the Parameterized_Access variant is used either:
  - to reference a COSEM object attribute to be read selectively. In this case, the Variable_Name element references the COSEM object attribute, the Selector and the Parameter elements carry the access selector and the access parameters respectively as specified in the attribute specification; or
  - to reference a COSEM object method to be invoked. In this case, the Variable_Name element references the method, the Selector element is zero and the Parameter element carries the method invocation parameters (if any) or null data;
  - the request may include one or more parameterized access parameters;

NOTE 1 With this, the Read service can transport information in both directions, just like the ACTION service used with LN referencing: method invocation parameters from the client to the server and return parameters from the server to the client.

- the Read_Data_Block_Access variant is used when one or more COSEM object methods are invoked and the encoded form of the request does not fit in a single APDU. The request may include a single Read_Data_Block_Access parameter. It carries block transfer control information and raw data:
  - the Last_Block element indicates if the given block is the last one (TRUE) or not (FALSE);
  - the Block_Number element carries the number of the actual block sent;

  - the Raw_Data element carries a part of the encoded form of the list of Variable_Access_Specification parameters (as it would be used without block transfer) including the method references and the method invocation parameters. Here, only the variants Variable_Name and Parameterized_Access are allowed.

- the Block_Number_Access variant is used when the server uses block transfer to send a long response, to confirm the reception of a data block and to request the next data block. The request may include a single Block_Number_Access parameter. It carries the number of the latest data block received correctly.

The Result (+) parameter indicates that the requested service has succeeded.

Without block transfer, the .response / .confirm service primitives contain one or more Read_Result parameters. Their number and order shall be the same as that of the Variable_Name / Parameterized_Access parameters in the .request / .indication primitives.

If the Read service is used to read attribute(s), then:

- the Data choice is taken to carry the value of the attribute at the time of access;
- the Data_Access_Error is taken to carry the reason for the read to fail for this attribute.

If the Read service is used to invoke method(s), then:

- the Data choice is taken to carry the return parameters (if data are returned, this implies that the method invocation succeeded). If there are no return parameters, Data shall be null data;

  NOTE 2 However, if no return data are expected, the Write service is used to invoke methods.

- the Data_Access_Error choice is taken to carry the reason for the method invocation to fail for this method.

In the case of block transfer, the .response / .confirm primitive contains a single Read_Result parameter. The Data_Block_Result choice is taken to carry one block of the response:

- the Last_Block element indicates whether the given block is the last one (TRUE) or not (FALSE);
- the Block_Number element shall carry the number of the block sent;
- the Raw_Data element contains a part of the encoded form of the list of Read_Results.

If the data block cannot be provided, then the .response primitive shall carry a single Result parameter using the Data_Access_Error choice, carrying an appropriate error message, for example (14) data-block-unavailable.

If the block number in the request is not the one expected, or if the next block cannot be delivered, then the Read.response service primitive shall be returned with a single Read_Result parameter, with the choice Data_Access_Error, carrying an appropriate code, for example (19) data-block-number-invalid.

The Block_Number choice is taken when the Read service is used to invoke one or more methods and the request is sent in several blocks, to confirm the correct reception of a data block and to ask for the next block. It carries the number of the latest block received.

The Result (–) parameter indicates that the service previously requested failed. The Error_Type parameter provides the reason for failure. In this case, the server shall send back a ConfirmedServiceError APDU instead of a ReadResponse APDU.

Use

A possible logical sequence of the Read service primitives is illustrated in Figure 35 item a).

The Read.request primitive is invoked following the invocation of a GET or ACTION .request primitive by the client AP and mapping this to a Read.request primitive by the SN_MAPPER ASE. The client AL builds then the ReadRequest APDU and sends it to the server. For LN / SN service mapping, see 6.19.

The Read.indication primitive is generated by the server AL upon reception of a ReadRequest APDU.

The Read.response primitive is invoked by the server AP in order to send a response to a previously received Read.indication primitive. The server AL builds then the ReadResponse APDU and sends it to the client.

The Read.confirm primitive is generated by the client AL following the reception of a ReadResponse APDU. It is then mapped back to a GET or ACTION .confirm primitive by the SN_MAPPER ASE and the GET or ACTION .confirm primitive is generated.

The protocol of the Read service is specified in 7.3.9.

6.15 The Write service

Function

The Write service is used with SN referencing. It is a confirmed service. Its functions are:

- to write the value of one or more COSEM interface object attributes;
- to invoke one or more COSEM interface object methods when no return parameters are expected.

In both cases, if the encoded form of the .request service primitive does not fit in a single APDU, then it can be sent in several requests with block transfer. The .response service primitive shall always fit in a single APDU.

The Write service is specified in IEC 61334-4-41:1996, 10.5 and Annex A. For completeness and for consistency with the specification of services using LN referencing, the specification is reproduced here, together with the extensions made for DLMS/COSEM.

Semantics of the service primitives

The Write service primitives shall provide service parameters as shown in Table 55.

Table 55 – Service parameters of the Write service

	.request	.indication	.response	.confirm
Variable_Access_Specification { Variable_Access_Specification }	M	M(=)	–	–
Variable_Name	S	S (=)		
Parameterized_Access	S	S (=)		
Variable_Name	M	M (=)		
Selector	M	M (=)		
Parameter	M	M (=)		
Write_Data_Block_Access	S	S (=)	–	–
Last_Block	M	M (=)		
Block_Number	M	M (=)		
Data { Data }	M	M (=)	–	–
Result (+)	–	–	S	S (=)
Write_Result { Write_Result }	-	-	S	S (=)
Success			S	S (=)
Data_Access_Error			S	S (=)
Block_Number			S	S (=)
Result (–)			S	S (=)
Error_Type			M	M (=)
NOTE For security parameters, see Table 40.				

The use of the different variants of the Variable-Access-Specification service parameter of the Write.request service primitive and the different choices of the Write.response primitive are shown in Table 56. The use of the Data service parameter is also explained.

If the encoded form of the request does not fit in a single APDU, it can be transported in data blocks using either the service-specific or the general block transfer mechanism.

Table 56 – Use of the Variable_Access_Specification variants and the Write.response choices

Write.request Variable_Access_Specification		Write.response CHOICE	
Variable_Name {Variable_Name}	References a list[1] of COSEM object attributes. The Data service parameter carries the data to be written or the method invocation parameter(s).	Success {Success}	Indicates that the attribute referenced could be successfully written.
		Data_Access_Error {Data_Access_Error}	Provides the reason for the write to fail.
Parameterized_Access {Parameterized_Access}	References a list[1] of COSEM object attributes to be written selectively. The Data service parameter carries the data to be written.	Success {Success}	As above.
		Data_Access_Error {Data_Access_Error}	
Write_Data_Block_Access	Carries block transfer control information. The Data service parameter carries raw-data, including the encoded form of the list[1] of COSEM object attribute or method references, and the list of data to be written or the list of method invocation parameters.	Block_Number	Carries the number of the latest data block received.
NOTE The same Write.response choice can be present more than once, to show the possible responses to each request. [1] A list may have one or more elements.			

The Write.request service primitive may have one or more Variable_Access_Specification parameters:

- the Variable_Name variant is used to reference a complete COSEM object attribute to be written or COSEM object method to be invoked. The request may include one or more variable names;
- the Parameterized_Access variant is used to reference a COSEM object attribute to be written selectively. In this case, the Variable_Name element references the COSEM object attribute, the Selector and the Parameter elements carry the access selector and the access parameters respectively as specified in the attribute specification. The request may include one or more Parameterized_Access parameters;

The Data service parameter carries the value(s) to be written to the attribute(s), or the method invocation parameter(s) of the method(s) to be invoked. The number and the order of the Data parameters shall be the same as that of the Variable_Access_Specification parameters.

If the Write.request service primitive does not fit into a single APDU, block transfer may be used. In this case:

- the Write_Data_Block_Access variant of the Variable_Access_Specification carries block transfer control information:
  - the Last_Block element indicates whether the given block is the last one (TRUE) or not (FALSE);
  - the Block_Number element carries the number of the actual block sent;
- the Data parameter carries one part of the list of the attribute references and the list of data to be written, or one part of the list of method references and the list of method invocation parameters.
- The request includes a single Write_Data_Block_Access and a single Data parameter.

The Result (+) parameter indicates that the service requested has succeeded.

The .response / .confirm service primitives contain a list of Write_Result parameters. Their number and order shall be the same as that of the Variable_Name / Parameterized_Access parameters in the .request / .indication service primitives.

Without block transfer, and with block transfer after receiving the last block:

- when the Write service is used to write attribute(s), each element carries either the success of the write access (Success) or a reason for the write to fail for this variable (Data_Access_Error);
- when the Write service is used to invoke method(s), each element carries either the success of the method invocation access (Success) or a reason for the method invocation to fail for this variable (Data_Access_Error).

The Block_Number choice is used during block transfer to confirm the correct reception of a data block and to ask for the next block. It carries the number of the latest block received.

If the block-number in the request is not the one expected, or if the block could not be received correctly, then the Write.response service primitive shall be returned with a single Write_Result parameter, with the choice Data_Access_Error, carrying an appropriate code, for example (19) data-block-number-invalid.

The Result (–) parameter indicates that the service requested has failed. The Error_Type parameter provides the reason for failure. In this case, the server shall send back a ConfirmedServiceError APDU instead of a WriteResponse APDU.

Use

A possible logical sequence of the Write service primitives is illustrated in Figure 35 item a).

The Write.request primitive is invoked following the invocation of a SET or ACTION .request primitive by the client AP and mapping this to a Write.request primitive by the SN_MAPPER ASE. The client AL builds then the WriteRequest APDU and sends it to the server. For LN / SN service mapping, see 6.19.

The Write.indication primitive is generated by the server AL upon reception of a WriteRequest APDU.

The Write.raesponse primitive is invoked by the server AP in order to send a response to a previously received Write.indication primitive. The server AL builds then the WriteResponse APDU and sends it to the client.

The Write.confirm primitive is generated by the client AL following the reception of a WriteResponse APDU. It is mapped then back to a SET or ACTION .confirm primitive by the SN_MAPPER ASE and the SET or ACTION .confirm primitive is generated.

The protocol of the Write service is specified in 7.3.10.

6.16 The UnconfirmedWrite service

Function

The UnconfirmedWrite service is used with SN referencing. It is an unconfirmed service. Its functions are:

- to write the value of one or more COSEM object attributes;
- to invoke one or more COSEM interface object method when no return parameters are expected.

The UnconfirmedWrite.request service primitive shall always fit in a single APDU.

The UnconfirmedWrite service is specified in IEC 61334-4-41:1996, 10.6 and Annex A. For completeness and for consistency with the specification of services using LN referencing, the specification is reproduced here, together with the extensions made for DLMS/COSEM.

Semantics of the service primitives

The UnconfirmedWrite service primitives shall provide service parameters as shown in Table 57.

Table 57 – Service parameters of the UnconfirmedWrite service

	.request	.indication
Variable_Access_Specification { Variable_Access_Specification }	M	M(=)
Variable_Name	S	S (=)
Parameterized_Access	S	S (=)
Variable_Name	M	M (=)
Selector	M	M (=)
Parameter	M	M (=)
Data { Data }	M	M (=)
NOTE For security parameters, see Table 40.		

The use of the different variants of the Variable-Access-Specification service parameter of the UnconfirmedWrite.request service primitive is shown in Table 58. The use of the Data service parameter is also explained.

If the encoded form of the request does not fit in a single APDU, it can be transported in data blocks using the general block transfer mechanism.

Table 58 – Use of the Variable_Access_Specification variants

UnconfirmedWrite.request Variable_Access_Specification	
Variable_Name {Variable_Name}	References a COSEM object attribute. The Data service parameter carries the data to be written or the method invocation parameter(s).
Parameterized_Access {Parameterized_Access}	References a COSEM object attribute with selective access. The Data service parameter carries the data to be written.

The UnconfirmedWrite.request service primitive may have one or more Variable_Access_Specification parameters.

- the Variable_Name variant is used to reference a complete COSEM object attribute to be written or COSEM object method to be invoked;
- the Parameterized_Access variant is used to reference a COSEM object attribute to be written selectively. In this case, the Variable_Name element references the COSEM object attribute, the Selector and the Parameter elements carry the access selector and the access parameters respectively as specified in the attribute specification.

The Data service parameter carries the value(s) to be written to the attribute(s), or the method invocation parameter(s) of the method(s) to be invoked. The number and the order of the Data parameters shall be the same as that of the Variable_Access_Specification parameters.

Use

A possible logical sequence of the Write service primitives is illustrated in Figure 35 item d).

The UnconfirmedWrite.request primitive is invoked following the invocation of a SET or ACTION .request primitive with Service_Class == Unconfirmed by the client AP and mapping this to an UnconfirmedWrite.request primitive by the SN_MAPPER ASE. The client AL builds then the UnconfirmedWriteRequest APDU and sends it to the server.

The UnconfirmedWrite.indication primitive is generated by the server AL upon reception of a WriteRequest APDU.

The protocol of the UnconfirmedWrite service is specified in 7.3.11.

6.17 The InformationReport service

Function

The InformationReport service is an unsolicited, non-client/server type service. It is requested by a server using SN referencing, upon occurrence of an event, in order to inform the client of the value of one or more DLMS named variables – mapped to COSEM interface object attributes – as though they had been requested by the client. It is an unconfirmed service.

The InformationReport service is specified in IEC 61334-4-41:1996, 10.7 and Annex A. For completeness and for consistency with the specification of services using LN referencing, the specification of the InformationReport service is reproduced here, together with the extensions made for DLMS/COSEM.

Semantics of the service primitives

The InformationReport service primitives shall provide parameters as shown in Table 59.

Table 59 – Service parameters of the InformationReport service

	.request	.indication
Current_Time	M	M (=)
Variable_Access_Specification { Variable_Access_Specification } Variable_Name	M M	M (=) M (=)
Data { Data }	M	M(=)

The Current_Time parameter indicates the time at which the InformationReport.request service primitive was issued.

The Variable_Access_Specification parameter of choice Variable_Name specifies one or more DLMS named variables – mapped to COSEM interface object attributes – the value of which is sent by the server.

The Data parameter carries the value of the DLMS named variable(s), in the same order as the order of the Variable_Access_Specification parameter(s).

The protocol for the InformationReport service is specified in 7.3.12.

Client side layer management services: the SetMapperTable.request

Function