Application Note 163/13 v01

Title: **Extended Interface Object addressing**

Status: **Date:**

Draft Proposal 2013.07.30

Transitional period: Immediate effect after Final Voting.

Date: 2013.07.30

Subject:

Documents

Modified

[01] Chapter 3/3/7 “Application Layer” v1.3.00 AS of 2010.10.22
[02] Chapter 3/5/1 “Resources”
[03] Chapter 3/5/2 “Management Procedures”
[04] Chapter 3/7/3 “Standard Identifier Tables”
[05] Chapter 3/4/1 “Application Interface Layer””

Referred

[06] Volume 6 “Profiles”
[07] Chapter 3/7/2 “Datapoint Types”

Document updates

Version	Date	Modifications
KSG511-01	2012.09.17	Initial Creation
KSG511-02	2012.11.28	Updated according the feedback of the KSG meeting of 2012.09.25 and 2012.09.26
KSG511-03	2013.02.05	Updated according the feedback of the KSG meeting of 2013.01.09 and the KSG online meeting of 2013.02.04
KSG511-04	2013.03.21	Updated according the feedback of the KSG meeting of 2013.03.06 and defining missing error codes
KSG511-05	2013.07.04	Updated according the feedback of the KSG meeting of 2013.06.12
KSG511-06	2013.07.10	Updated according e-mail feedback of Mr De Bruyne (05.07.2013)
AN163 v01	2013.07.30	Preparation of the Draft Proposal.

Contents

1 Purpose, motivation and scope

Currently only 255 Interface Objects can be used in a device. For bigger devices this is not enough if they use the Functional Block approach.

2 Specification

2.1 Terms and definitions

This clause is not intended for integration in the KNX Specifications.

This document does not introduce neither modify any terms or definitions.

2.2 Stack and communication

2.2.1 Application Layer

2.2.1.1 APDU

Insert following rows in table 1 in [05].

octet 6		octet 7									Use
1	0	7	6	5	4	3	2	1	0	...	
0	1	1	1	0	0	1	1	0	0		A_PropertyExtValue_Read-PDU
0	1	1	1	0	0	1	1	0	1		A_PropertyExtValue_Response-PDU
0	1	1	1	0	0	1	1	1	0		A_PropertyExtValue_WriteCon-PDU
0	1	1	1	0	0	1	1	1	1		A_PropertyExtValue_WriteConRes-PDU
0	1	1	1	0	1	0	0	0	0		A_PropertyExtValue_WriteUnCon-PDU
0	1	1	1	0	1	0	0	0	1		A_PropertyExtValue_InfoReport-PDU
0	1	1	1	0	1	0	0	1	0		A_PropertyExtDescription_Read-PDU
0	1	1	1	0	1	0	0	1	1		A_PropertyExtDescription_Response-PDU
0	1	1	1	0	1	0	1	0	0		A_FunctionPropertyExtCommand-PDU
0	1	1	1	0	1	0	1	0	1		A_FunctionPropertyExtState_Read-PDU
0	1	1	1	0	1	0	1	1	0		A_FunctionPropertyExtState_Response-PDU

2.2.1.2 Data Property Extended services

2.2.1.2.1 A_PropertyExtValue_Read

The A_PropertyExtValue_Read.req primitive shall be applied by the user of Application Layer to read the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The Interface Object of the partner shall be addressed with an Interface Object Type and the Object Instance and the Property of the Interface Object shall be addressed with a Property Identifier. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to read. The user of Application Layer in the partner device shall respond with an A_PropertyExtValue_Read.res, this is, the service shall be confirmed by the remote application process.

The local Application Layer shall accept the service request and pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters ASAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_Read-PDU to an A_PropertyExtValue_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_Read.ind primitive.

The application process shall respond to the A_PropertyExtValue_Read.ind primitive with an A_PropertyExtValue_Read.res primitive containing the requested number of elements of the Property value of the Property of the associated Interface Object.

NOTE 1 For the Object Instance more than one octet is required.

EXAMPLE 1 For a 4-channel DALI-gateway.

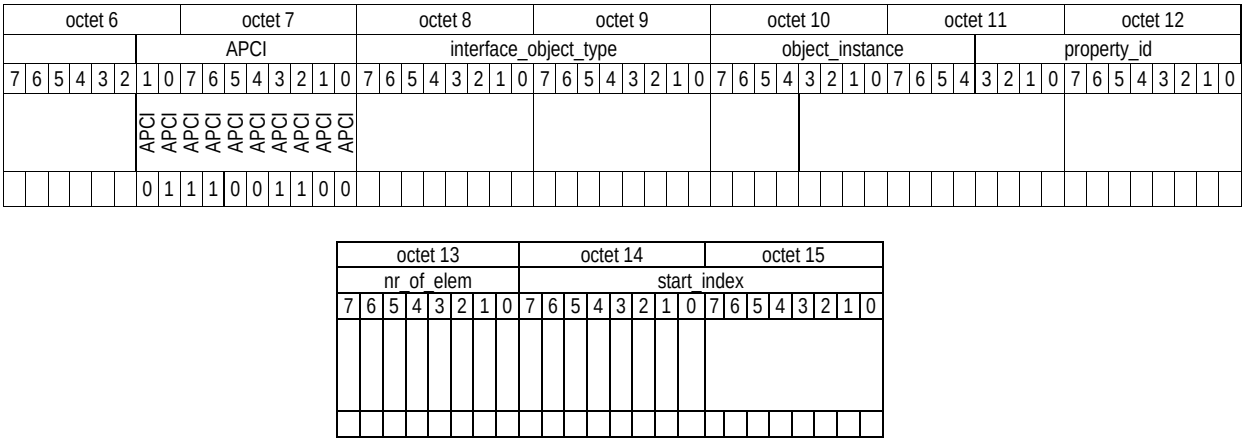


Figure 1 - A_PropertyExtValue_Read-PDU (example)

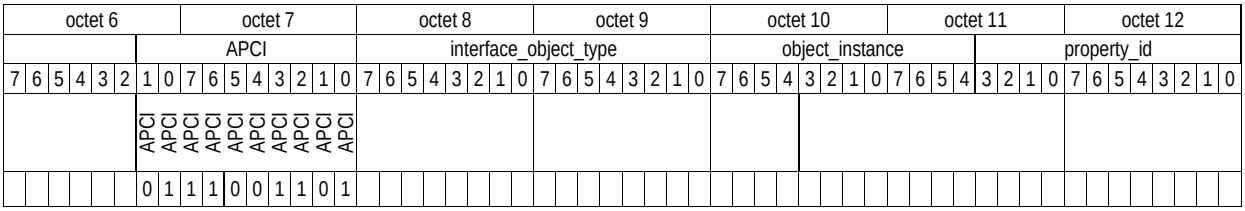


Figure 2 - A_PropertyExtValue_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_Response-PDU to an A_PropertyExtValue_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_Read.Acon primitive.

Error and exception handling

- If the MaS has a problem, e.g. Interface Object or Property does not exist, then the nr_of_elem shall be set to zero and the start_index of the response shall be set to same value as received with the request. The data field of a negative response shall contain error information. The error information of a negative confirmation shall be a one octet long enumerated data field. For more details concerning the error handling and the Error Code Set please refer to clause 2.2.1.2.6 “Error Codes”.

- If the remote application process receives an A_PropertyExtValue_Read-PDU with start_index = 0, this is, reading the current number of elements of the Property Value array, but with a nr_of_elem greater than 1, then it shall respond with an A_Property-ExtValue_Response-PDU with start_index = 0 and nr_of_elem = 1 and the field data shall contain the current number of elements of the Property array (2 octet value).

A_PropertyExtValue_Read.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:		the number of array elements to be read in the Property value
start_index:		the array index of the first array element to be read

A_PropertyExtValue_Read.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:		the number of array elements to be read in the Property value
start_index:		the array index of the first array element to be read
a_status:	ok:	A_PropertyExtValue_Read sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyExtValue_Read.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be read in the Property value
start_index:	the array index of the first array element to be read

A_PropertyExtValue_Read.res(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be read in the Property value
start_index:	the array index of the first array element to be read
data:	the value of the array elements read, or an error code if an error occurred

A_PropertyExtValue_Read.Acon(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be read in the Property value
start_index:	the array index of the first array element to be read
data:	the value of the array elements read, or an error code

2.2.1.2.2 A_PropertyExtValue_WriteCon-service

The A_PropertyExtValue_WriteCon.req primitive shall be applied by the user of Application Layer to modify the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The Interface Object of the partner shall be addressed with the Interface Object Type and the Object Instance and the Property of the Interface Object shall be addressed with the Property Identifier. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to write to.

The local Application Layer shall accept the service request and pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_WriteCon-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_WriteCon-PDU to an A_PropertyExtValue_WriteCon.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_WriteCon.ind primitive.

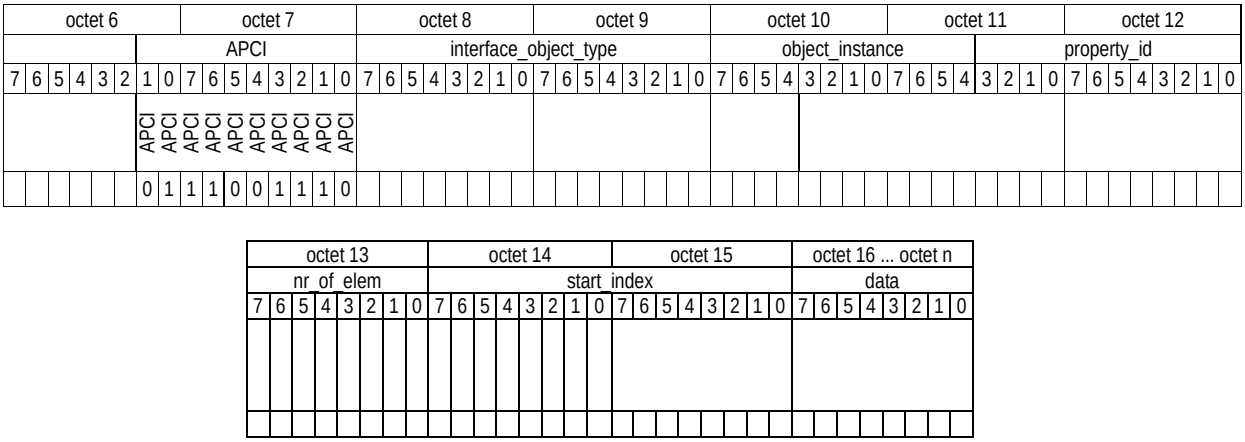


Figure 3 - A_PropertyExtValue_WriteCon-PDU (example)

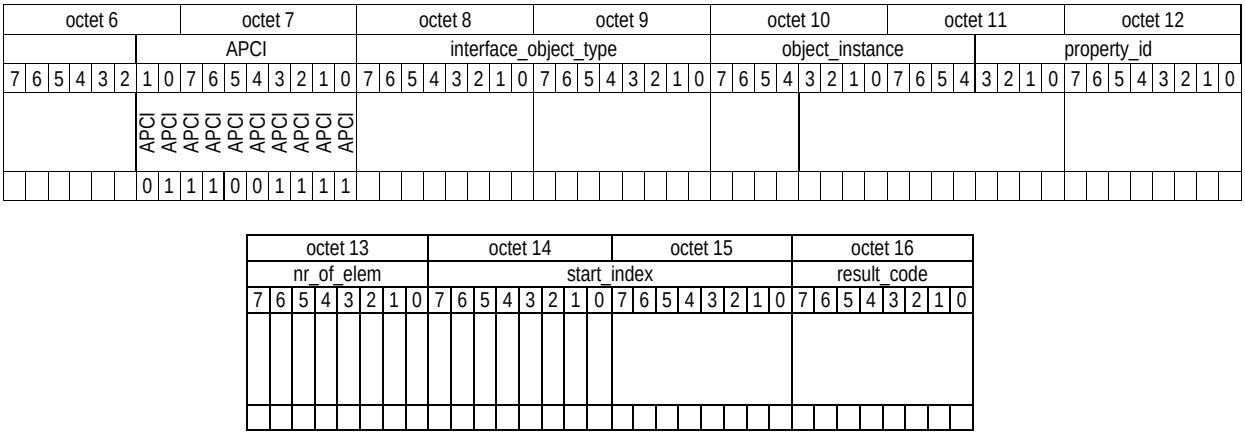


Figure 4 - A_PropertyExtValue_WriteConRes-PDU (example)

The remote application process shall respond to the A_PropertyExtValue_WriteCon.ind primitive with an A_PropertyExtValue_WriteCon.res primitive containing the requested number of elements of the Property value of the Property of the associated Interface Object. The result_code shall be zero if the write operation is successful, otherwise the result_code shall contain the appropriate Error Code as specified in clause 2.2.1.2.6 “Error Codes”.

Error and exception handling

- If the MaS has a problem, e.g. Interface Object or Property does not exist, then the nr_of_elem shall be set to zero and the Start Index of the response shall be set to same value as received with the request. The result_code field of a negative response shall contain error information. The error information of a negative confirmation shall be a one octet long enumerated data field. For more details concerning the error handling and the Error Code Set please refer to clause 2.2.1.2.6 “Error Codes”.

The remote Application Layer shall accept the service response and shall pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_WriteConRes-PDU.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_WriteConRes-PDU to an A_PropertyExtValue_WriteCon.Acon primitive if an A_PropertyExtValue_WriteCon-PDU is sent before to this communication partner to this Interface Object and Property. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_WriteCon.Acon primitive.

A_PropertyExtValue_WriteCon.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
data:	the data to write to the array elements

A_PropertyExtValue_WriteCon.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:		the number of array elements to be written in the Property value
start_index:		the array index of the first array element to be written
data:		the data to write to the array elements
a_status:	ok:	A_PropertyExtValue_WriteCon sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyExtValue_WriteCon.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
data:	the data to write to the array elements

A_PropertyExtValue_WriteCon.res(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_result)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
a_result:	the result code of the operation

A_PropertyExtValue_WriteCon.Acon(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_result)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
a_result:	the result code of the operation

2.2.1.2.3 A_PropertyExtValue_WriteUnCon-service

The A_PropertyExtValue_WriteUnCon.req primitive shall be applied by the user of Application Layer to modify the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The Interface Object of the partner shall be addressed with the Interface Object Type and the Object Instance and the Property of the Interface Object shall be addressed with the Property Identifier. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to write to.

The local Application Layer shall accept the service request and pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_WriteUnCon-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_WriteUnCon-PDU to an A_PropertyExtValue_WriteUnCon.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_WriteUnCon.ind primitive.

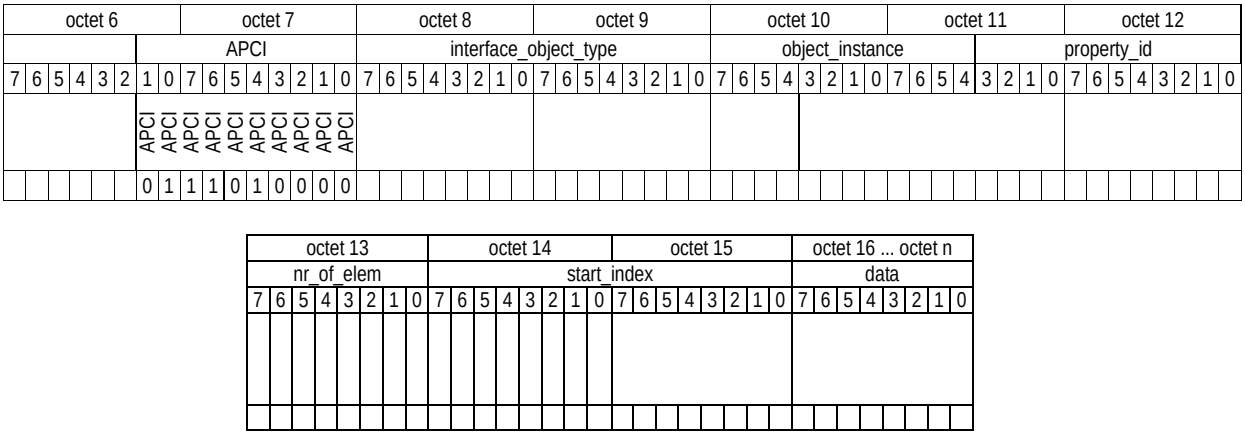


Figure 5 - A_PropertyExtValue_WriteUnCon-PDU (example)

The remote application process shall not respond to the A_PropertyExtValue_WriteUnCon.ind primitive.

Error and exception handling

- If the MaS has a problem - e.g. Interface Object or Property does not exist - then the request shall be ignored.

A_PropertyExtValue_WriteUnCon.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
data:	the data to write to the array elements

A_PropertyExtValue_WriteUnCon.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:		the number of array elements to be written in the Property value
start_index:		the array index of the first array element to be written
data:		the data to write to the array elements
a_status:	ok:	A_PropertyExtValue_WriteUnCon sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyExtValue_WriteUnCon.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be written in the Property value
start_index:	the array index of the first array element to be written
data:	the data to write to the array elements

2.2.1.2.4 A_PropertyExtValue_InfoReport-service

The A_PropertyExtValue_InfoReport.req primitive shall be applied by the user of Application Layer to report the value of a Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The value reported shall be addressed with its Interface Object Type and its Object Instance, and the Property of the Interface Object shall be addressed with the Property Identifier. The nr_of_elem and start_index shall indicate the number of array elements starting with the given start_index in the Property value that the user wants to report.

The local Application Layer shall accept the service request and pass it with a T_Data_-Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtValue_InfoReport-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtValue_InfoReport-PDU to an A_PropertyExtValue_InfoReport.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtValue_InfoReport.ind primitive.

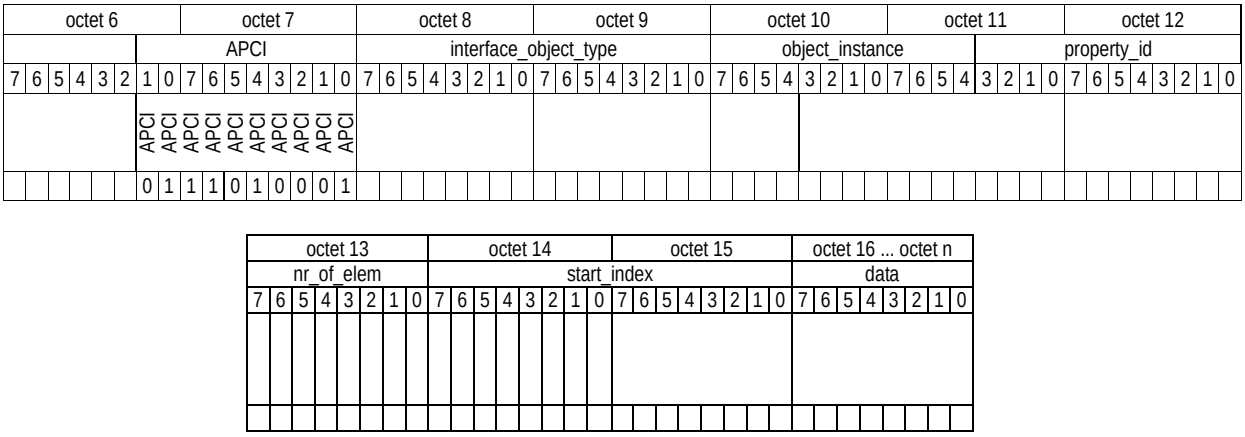


Figure 6 - A_PropertyExtValue_InfoReport-PDU (example)

The remote application process shall not respond to the A_PropertyExtValue_InfoReport.ind primitive.

Error and exception handling

- The application program ignores the data if the device is not a consumer of the Property.

A_PropertyExtValue_InfoReport.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be reported in the Property value
start_index:	the array index of the first array element to be reported
data:	the data to report

A_PropertyExtValue_InfoReport.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:		the number of array elements to be reported in the Property value
start_index:		the array index of the first array element to be reported
data:		the data to report
a_status:	ok:	A_PropertyExtValue_InfoReport sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyExtValue_InfoReport.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, nr_of_elem, start_index, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
nr_of_elem:	the number of array elements to be reported in the Property value
start_index:	the array index of the first array element to be reported
data:	the data to report

2.2.1.2.5 A_PropertyExtDescription_Read-service

Object enumeration is done via the property "IO List" in the device object.

The A_PropertyExtDescription_Read.req primitive shall be applied by the user of Application Layer, to read the description of the Property of an Interface Object. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the Transport Layer. The Interface Object of the partner shall be addressed with an Interface Object Type and an Object Instance and the Property of the Interface Object shall be addressed with a Property Identifier or with a Property Index. The Property Index shall be used only if the Property Identifier is zero. The Type shall be zero.

The Property Index, if evaluated, shall address the Property of the Interface Object with a sequential number: Property Index = 0 shall mean the first Property of the associated Interface Object, Property Index = 1 shall mean the second Property.

The remote application process shall confirm the service.

The local Application Layer shall accept the service request and shall pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an A_PropertyExtDescription_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtDescription_Read-PDU to an A_PropertyExtDescription_Read.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtDescription_Read.ind primitive.

The application process shall respond to the A_PropertyExtDescription_Read.ind primitive with an A_PropertyExtDescription_Read.res primitive containing the description of the addressed Property. The fields of the description of the Property shall be encoded as defined in [05].

- If the Property Identifier in the A_PropertyExtDescription_Read-PDU is zero then the remote application process shall use the indicated Property Index to access the Property description.
- If the Property Identifier in the A_PropertyExtDescription_Read-PDU is not zero, then the field Property Index shall be ignored (index should be zero) and the remote application process shall use the indicated Property Identifier to access the Property description.

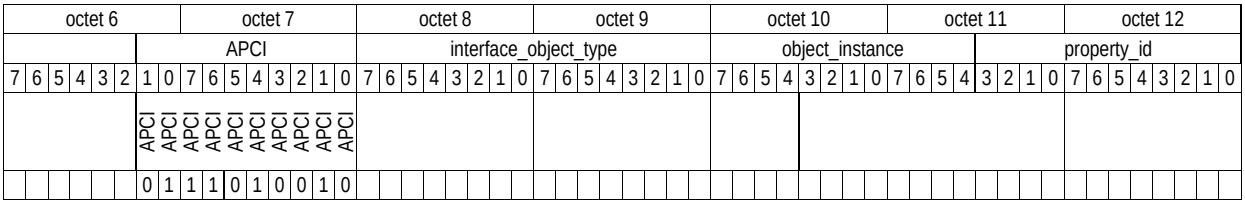
The Property Index in the A_PropertyExtDescription_Response-PDU shall be the correct value of the Property Index of the addressed Property

The Property Index shall contain the index of the addressed Property.

Error - and exception handling

- If the remote application process has a problem - e.g. Interface Object or Property does not exist - then the max_nr_of_elem of the A_PropertyExtDescription_Response-PDU shall be zero.

The A_PropertyExtDescription_Read-service shall not be subject to authorization. The remote application process shall not confirm the service negatively for authorization reasons (see A_Authorize_Request-service).



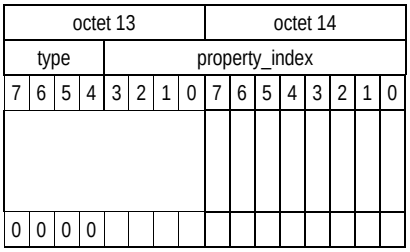


Figure 7 - A_PropertyExtDescription_Read-PDU (example)

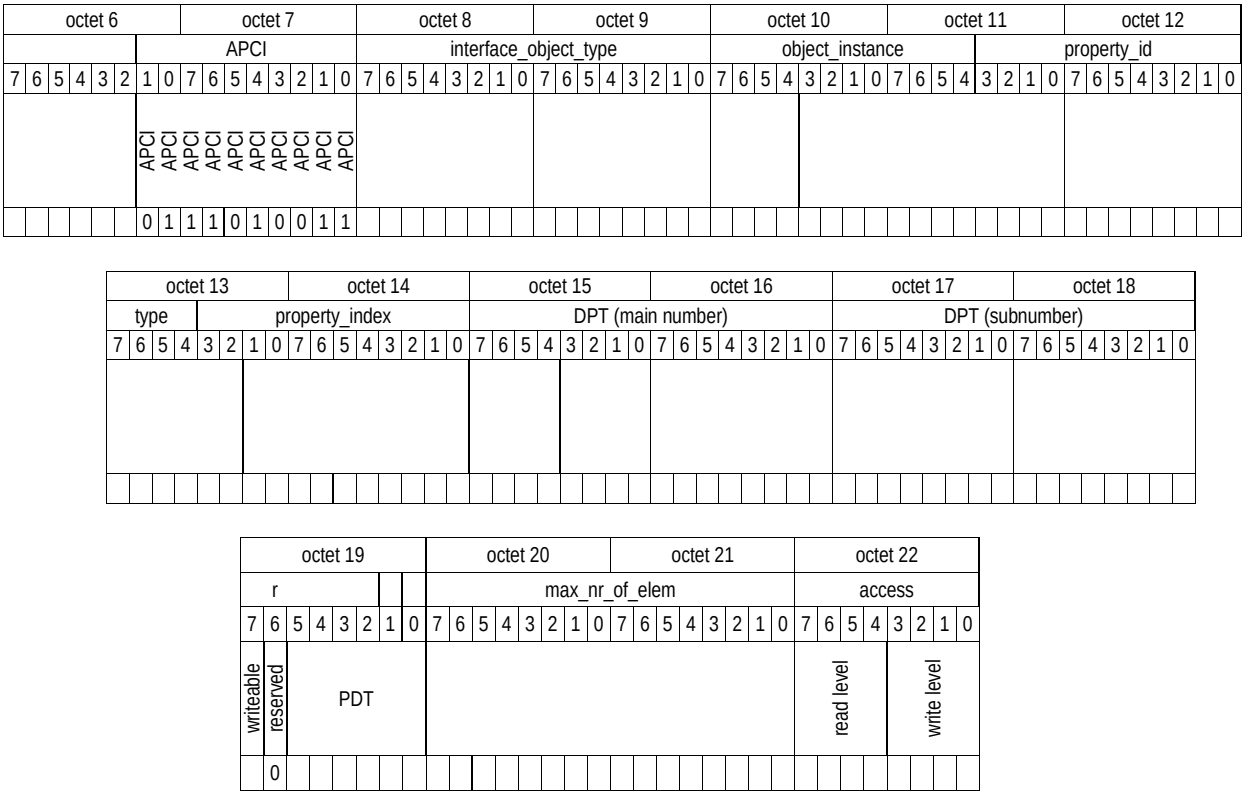


Figure 8 - A_PropertyExtDescription_Response-PDU (example)

The remote Application Layer shall accept the service response and shall pass it with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive; the TSDU shall be an A_PropertyExtDescription_Response-PDU.

Unused fields in the A_PropertyExtDescription_Response-PDU are marked with ‘r’ in Figure 8 above and shall be set to 0.

The local Application Layer shall map a T_Data_Individual.ind primitive with TSDU = A_PropertyExtDescription_Response-PDU to an A_PropertyExtDescription_Read.Acon primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_PropertyExtDescription_Read.Acon primitive.

Error - and exception handling

- If the MaS does not support the requested Type then it shall answer with Type zero.

A_PropertyExtDescription_Read.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, property_index, type)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
property_index:	sequential Property number
type:	the Type of the Property description, currently only type 0 defined

A_PropertyExtDescription_Read.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, property_index, type, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
property_index:		sequential Property number
type:		the Type of the Property description, currently only type 0 defined
a_status:	ok:	A_PropertyExtDescription_Read sent successfully with T_Data_Individual service
	not_ok:	transmission of the associated T_Data_Individual request frame did not succeed

A_PropertyExtDescription_Read.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, property_index, type)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
property_index:	sequential Property number
type:	the Type of the Property description, currently only type 0 defined

A_PropertyExtDescription_Read.res(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, property_index, type, dpt, write_enable, pdt, max_nr_of_elem, access)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
property_index:	sequential Property number
type:	the Type of the Property description, currently only type 0 defined
dpt:	the Datapoint Type according to which the Property Value is encoded
write_enable:	specifies whether the Property value can be written or not
pdt:	the Property Datatype according to which the Property value is encoded
max_nr_of_elem:	maximum number of elements of the array or zero to indicate a problem
access:	access level to read or write to the Property value

A_PropertyExtDescription_Read.Acon(priority, hop_count_type, ASAP, object_type, object_instance, property_id, property_index, type, DPT, write_enable, PDT, max_nr_of_elem, access)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
property_index:	sequential Property number
type:	the Type of the Property description, currently only type 0 defined
DPT:	the Datapoint Type according to which the Property Value is encoded
write_enable:	specifies whether the Property value can be written or not
PDT:	the Property Datatype according to which the Property value is encoded
max_nr_of_elem:	maximum number of elements of the array or zero to indicate a problem
access:	access level to read or write to the Property value

2.2.1.2.6 Error Codes

Error Code	ErrorType	Description
00h	No Error	
01h	Out of Range	write value not allowed (general, if not error 2 or 3)
02h	Out of MaxRange	write value too high
03h	Out of MinRange	write value too low
04h	Memory Error	memory cannot be written or only with fault(s)

Error Code	ErrorType	Description
05h	Read Only	write access to a “read only” or a write protected Property
06h	Illegal COMMAND	COMMAND not valid or not supported (for PDT_COMMAND)
07h	Void DP	read or write access to an non-existing Property
08h	Type Conflict	write access with a wrong datatype (Datapoint length)
09h	Prop. Index Range Error	read or write access to a non-existing Property array index (read acces: index ≥ number_of_elements; write access index ≥ max_number_of_elements)

2.2.1.3 Function Property Extended services

2.2.1.3.1 A_FunctionPropertyExtCommand-Service

The A_FunctionPropertyExtCommand.req primitive shall be applied by the user of Application Layer to call a Function Property of an Interface Object in a communication partner. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the local Transport Layer. The Interface Object of the partner shall be addressed with the Interface Object Type and the Object Instance and the Function Property of the Interface Object shall be addressed with the Property Identifier.

The local Application Layer shall accept the service request and pass it with a T_Data_Individual.req or a T_Data_Connected.req (depending whether the service is called connectionless or connection-oriented) to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req or the T_Data_Connected.req primitive; the TSDU shall be an A_FunctionPropertyExtCommand-PDU.

The remote Application Layer shall map a T_Data_Individual.ind or a T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyExtCommand-PDU to an A_FunctionPropertyExtCommand.ind primitive. The parameter comm_mode shall be set to “connection_oriented” in case of a T_Data_Connected.ind and to “connectionless” in case of a T_Data_Individual.ind primitive. The arguments TSAP and priority shall be mapped to the corresponding arguments ASAP and priority of the A_FunctionPropertyExtCommand.ind primitive.

The remote application shall check whether the Property Datatype of the Property addressed by this A_FunctionPropertyExtCommand.ind is PDT_Function. If this is the case it shall call the function and pass the input parameters to it, otherwise the error handling shall apply; see clause 0.

The remote application process shall respond to the A_FunctionPropertyExtCommand.ind primitive with an A_FunctionPropertyExtCommand.res primitive containing the return_code and the function specific output data.

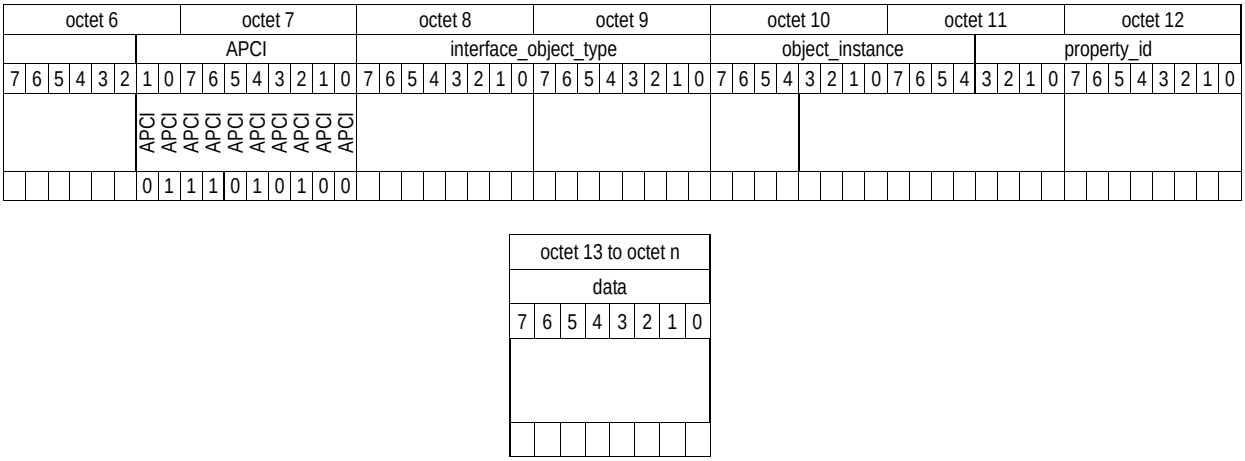


Figure 9 - A_FunctionPropertyExtCommand-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_-Individual.req or a T_Data_Connected.req (depending whether the service shall be sent connection-oriented or connectionless) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyExtState_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind or T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyExtState_Response-PDU to an A_FunctionPropertyExtCommand.Acon primitive.

A_FunctionPropertyExtCommand.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
data:	input data to the function

A_FunctionPropertyExtCommand.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
data:		input data to the function
a_status:	ok:	A_FunctionPropertyExtCommand sent successfully with T_Data_Individual- or T_Data_Connected-service
	not_ok:	transmission of the associated T_Data_Individual – or T_Data_Connected - request Frame did not succeed

A_FunctionPropertyExtCommand.Ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
data:	input data to the function

A_FunctionPropertyExtCommand.res(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, return_code, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
return_code:	return code returned by the function
data:	Output data from the function

A_FunctionPropertyExtCommand.Acon(priority, hop_count_type, ASAP, object_type, object_instance, property_id, return_code, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
return_code	return code returned by the function
data:	Output data from the function

2.2.1.3.2 A_FunctionPropertyExtState_Read-service

The A_FunctionPropertyExtState_Read.req primitive shall be applied by the user of Application Layer to read the state of a Function Property of an Interface Object in a remote device. The communication partner shall be addressed with a local ASAP that shall be mapped to an Individual Address by the local Transport Layer. The Interface Object of the remote device shall be addressed with an Interface Object Type and the Object Instance and the Property of the Interface Object shall be addressed with a Property Identifier. The user of the Application Layer in the remote device shall respond with an A_FunctionPropertyExt-State_Read.res; this is, the remote application process service shall confirm the service.

The local Application Layer shall accept the service request and pass it with a T_Data_-Individual.req or a T_Data_Connected.req (depending whether the service is sent connectionless or connection-oriented) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyExtState_Read-PDU.

The remote Application Layer shall map a T_Data_Individual.ind primitive or a T_Data_-Connected.ind primitive with TSDU = A_FunctionPropertyExtState_Read-PDU to an A_FunctionPropertyExtState_Read.ind primitive.

The remote application shall check whether the Property Datatype of the Property addressed by this A_FunctionPropertyExtState_Read.ind is PDT_Function. If this is the case it shall call the function and pass the input parameters to it, otherwise the error handling shall apply; see clause 0.

The application process shall respond to the A_FunctionPropertyExtState_Read.ind primitive with an A_FunctionPropertyExtState_Read.res primitive containing the function specific Return Code and the function specific output data.

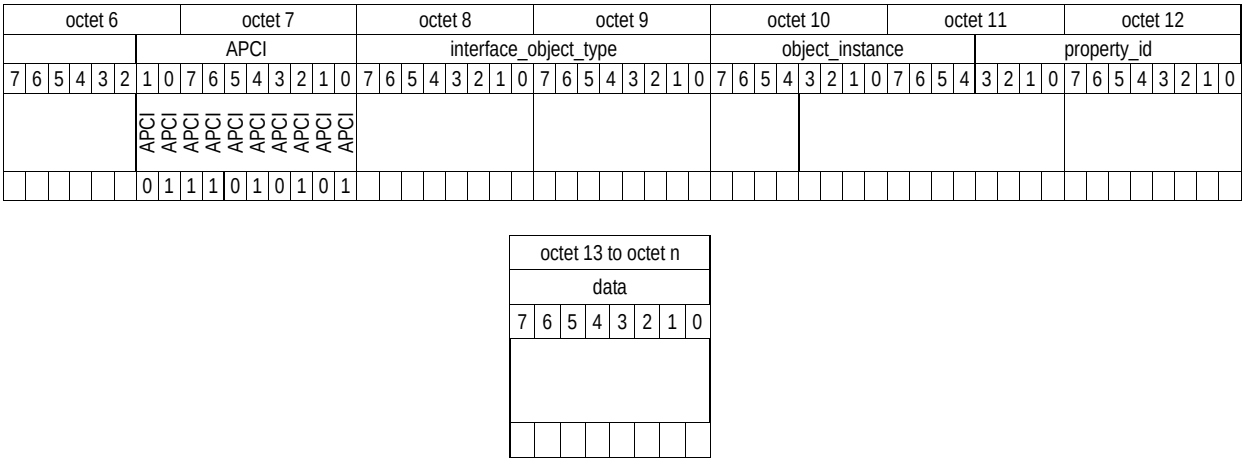


Figure 10 - A_FunctionPropertyExtState_Read-PDU (example)

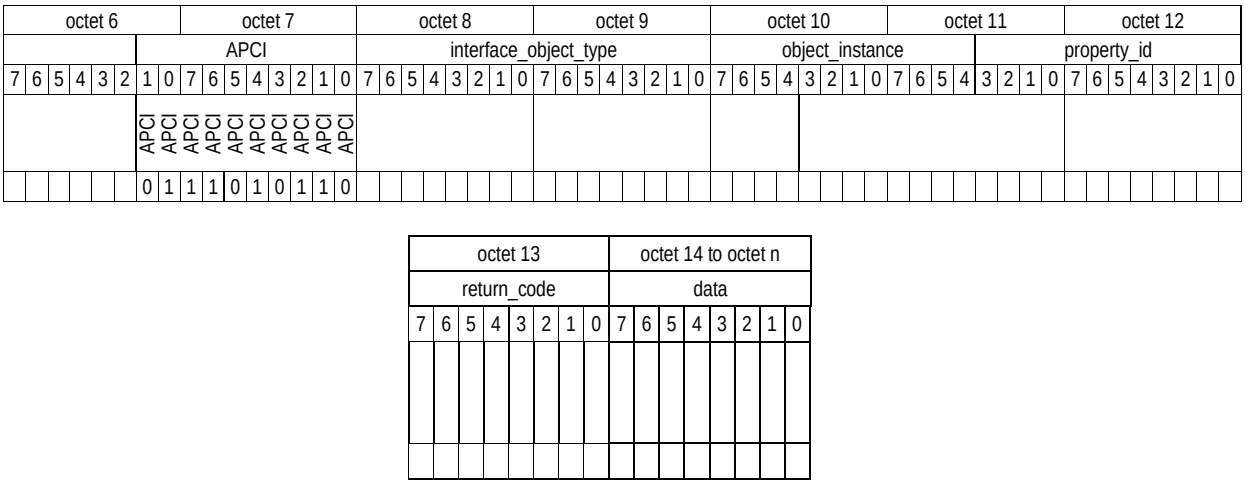


Figure 11 - A_FunctionPropertyExtState_Response-PDU (example)

The remote Application Layer shall accept the service response and pass it with a T_Data_-Individual.req or a T_Data_Connected.req (depending whether the service shall be sent connectionless or connection-oriented) to the local Transport Layer. The TSDU shall be an A_FunctionPropertyExtState_Response-PDU.

The local Application Layer shall map a T_Data_Individual.ind or T_Data_Connected.ind primitive with TSDU = A_FunctionPropertyExtState_Response-PDU to an A_Function-PropertyExtState_Read.Acon primitive.

A_FunctionPropertyExtState_Read.req(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, data)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
data:		input data to the function for reading

A_FunctionPropertyExtState_Read.Lcon(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, data, a_status)

ack_request:		Data Link Layer Acknowledge requested or don't care
priority:		system, urgent, normal or low priority
hop_count_type:		hop count 7 or standard
ASAP:		local reference of the Service Access Point or Individual Address
object_type:		the Interface Object Type of the addressed Interface Object
object_instance:		the Object Instance of the addressed Interface Object
property_id:		the Property Identifier of the Property of the addressed Interface Object
data:		input data to the function for reading
a_status:	ok:	A_FunctionPropertyExtState_Read-PDU sent successfully with T_Data_Individual - or T_Data_Connected service
	not_ok:	transmission of the associated T_Data_Individual – or T_Data_Connected request Frame did not succeed

A_FunctionPropertyExtState_Read.ind(priority, hop_count_type, ASAP, object_type, object_instance, property_id, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
data:	input data to the function for reading

A_FunctionPropertyExtState_Read.res(ack_request, priority, hop_count_type, ASAP, object_type, object_instance, property_id, return_code, data)

ack_request:	Data Link Layer Acknowledge requested or don't care
priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
return_code	error code returned by the function
data:	output data from the function

A_FunctionPropertyExtState_Read.Acon(priority, hop_count_type, ASAP, object_type, object_instance, property_id, return_code, data)

priority:	system, urgent, normal or low priority
hop_count_type:	hop count 7 or standard
ASAP:	local reference of the Service Access Point or Individual Address
object_type:	the Interface Object Type of the addressed Interface Object
object_instance:	the Object Instance of the addressed Interface Object
property_id:	the Property Identifier of the Property of the addressed Interface Object
return_code	error code returned by the function
data:	Output data from the function

2.2.1.3.3 Error and exception handling for Function Property services

- If the Interface Object Property accessed by A_FunctionPropertyExtCommand-PDU or by A_FunctionPropertyExtState_Read-PDU is not of the Property Datatype PDT_Function then the remote application shall respond with a A_FunctionPropertyExtState_Response-PDU without the field return_code (this is, the returned PDU shall not contain the field return_code) and without the field data (this is, the returned PDU shall not contain the field data).
- In case the remote application is able to successfully call a Function Property, then the Function Property shall deliver a return_code in the field return_code of the PDU. The following rules shall apply for all functions.
  - Return_code = 00h: function successfully executed, this is the return code 00h shall be the positive result of the function.
  - Return_code ≠ 00h: error.
    Error codes are defined in a function specific way.
- In case an Interface Object Property of Property Datatype PDT_Function is accessed via the A_PropertyExtValue_Read-service or the A_PropertyExtValue_Write-service then the Application Layer shall respond with an A_PropertyExtValue_Response-PDU with the standard error handling for Data Properties (this is the field nr_of_elem shall be zero and there shall be no data field).
- In case an Interface Object Property of Property Datatype PDT_Function is accessed via the A_PropertyExtDescription_Read, the Application Layer shall respond with an A_PropertyExtDescription_Response-PDU with type = PDT_Function and max_nr_of_elem = 1; read_level and write_level can have any value.

2.2.1.4 A_ADC_Read-service

In the Application Layer specification ([01]) clause 3.5.2 "A_ADC_read-service" Figure 46 "A_ADC_Read-PDU" shall be changed as follows.

octet 6								octet 7								octet 8							
								APCI							Channel_nr	Read_count							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
								0	1	1	0	0	0	0									

In the Application Layer specification ([01]) clause 3.5.2 "A_ADC_read-service" Figure 47 "A_ADC_Response-PDU" shall be changed as follows.

Octet 6								Octet 7								Octet 8								Octet 9								Octet 10							
								APCI							Channel_nr	Read_count								Sum															
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
																								Value high								Value low							
						0	1	1	0	0	0	0																											

2.2.2 Application Interface Layer

2.2.2.1 Interface Object Server

2.2.2.1.1 Common Structure

4.1 shall be changed as followed:

Interface Objects shall be instances of a common general structure. Each Interface Object Instance in a device shall have a unique identifier in the device, the Object Index.

Each Interface Object in a device shall be addressed by an Object Index or by an Interface Object Type and an Interface Object Instance. The Object Index shall be unique within the device. Interface objects of the same Interface Object Type shall be numbered by the Interface Object Instance starting with 1. So the combination of Interface Object Type and Interface Object Instance shall be unique. Each Property of an Interface Object shall be addressed with a Property Identifier. The Property Identifier shall be unique for the Interface Object.

Properties with a Property Identifier ≤ 255 shall have a lower Property Index than all properties with a Property Index > 255 as the Properties with a Property Identifier > 255 are only accessible with extended Property Services.

For the services A_PropertyDescription_Read and A_PropertyExtDescription_Read, a Property may be addressed also by the Property Index.

Each Interface Object shall at least contain the Property Object Type.

All Interface Object shall base on the common structure as presented in Figure 12. Depending on the flavour of the Interface Object, part of this structure may not be present.

Device				
	Interface Object			
		Property		
			Property description	
				Property Identifier (2 octet, unsigned12) = PID_OBJECT_TYPE
				Property Datatype (1 octet, unsigned8)
				DPT (4 octet, unsigned16 unsigned16)
				max_nr_of_elem (unsigned 16)
				Access (1 octet, unsigned4unsigned4)
			Property value	
				Array(0)=current nr. of elements (unsigned 16)
				Array(1 … max_nr_of_elem) = value
		…		
		Property		
			Property description	
				Property Identifier (2 octet, unsigned12)
				Property Datatype (1 octet, unsigned8)
				DPT (4 octet, unsigned16 unsigned16)
				max_nr_of_elem (unsigned 16)
				Access (1 octet, unsigned4unsigned4)
			Property value	
				Array(0)=current nr. of elements (unsigned 16)
				Array(1 … max_nr_of_elem) = value
		…		
		Property		
			Property description	
				Property Identifier (2 octet, unsigned12)
				Property Datatype (1 octet, unsigned8)
				DPT (4 octet, unsigned16 unsigned16)
				max_nr_of_elem (unsigned 16)
				Access (1 octet, unsigned4unsigned4)
			Property value	
				Array(0)=current nr. of elements (unsigned 16)
				Array(1 … max_nr_of_elem) = value

Figure 12 - Interface Object structure

From this common structure of Interface Objects, three types are derived.

1. Full Interface Objects, and
2. Reduced Interface Objects, and .
3. Extended Interface Objects.

These types are specified in the following clauses.

NOTE 2 If in the following specification the specific type is not explicitly mentioned or clear from the scope, then the requirements apply for both Full Interface Objects as well as for Reduced Interface Objects.

The Extended Interface Objects are an extension of Full Interface Objects. If a single device supports extended Interface Objects then it shall support this for all Interface Objects. There shall be no co-existence of other types.

If no Extended Interface Objects are supported then in a single device both other types of Interface Objects (Full - and Reduced) may co-exist.

This means that the following cases are possible.

1. A device contains only Reduced Interface Objects.
2. A device contains both Reduced - and Full Interface Objects. This means that the service A_PropertyDescription_Read shall be implemented in this device, but is only available for the Full Interface Objects.
3. A device contains only Full Interface Objects.
4. A device contains only Extended Interface Objects but all Access method as defined in Full Interface Objects also possible.

In [06] it is documented which combinations of Full - and Reduced Interface Objects are allowed.

2.2.2.1.2 Types of Interface Objects

2.2.2.1.2.1 Overview

4.3 Insert overview over different types

Feature	Reduced Interface Objects	Full Interface Objects	Extended Interface Objects
Object			
Object Index	U_8	U_8	X
Property description			
Property Identifier	U_8	U_8	U_{12}
Property Index	X	**U_8**	U_{12}
Write enable flag	X	M	M
Property Datatype	X	M	M
Property Datapoint Type	X	O	M
read- and write access levels	X	M	M
Property Value			
minimal Property (array element) value size (octets)	-	?	?
maximal Property (array element) value size (octets)	10	?	?
max_nr_of_elem	X	U_{12}	U_{16}
current_nr_of_elem	U_{16}	U_{16} U_{32}	U_{16}
nr_of_elem (in services)	U_4	U_4	U_8
start_index (in services)	U_{12}	U_{12}	U_{16}
Property services			
A_PropertyDescription_Read	X	M	M

Feature	Reduced Interface Objects	Full Interface Objects	Extended Interface Objects
A_PropertyValue_Read	M	M	M
A_PropertyValue_Write	M	M	M
A_FunctionPropertyCommand	X	O	O
A_FunctionPropertyState_Read	X	O	O
Extended Property Services			
A_PropertyExtDescription_Read	X	O	M
A_PropertyExtValue_Read	X	O	M
A_PropertyExtValue_WriteCon	X	O	M
A_PropertyExtValue_WriteUnCon	X	O	M
A_PropertyExtValue_InfoReport	X	O	O
A_FunctionPropertyExtCommand	X	O	O
A_FunctionPropertyExtState_Read	X	O	O

2.2.2.1.2.2 Full Interface Object

2.2.2.1.2.2.1 Definition

4.3.1.1 Change text of this paragraph to following text:

Full Interface Objects shall comply with the common data structure as specified in Figure 12 with the exception that the DPT is optional. Every Interface Object shall consist of a number of Properties. Every Property shall consist of

- one Property Description and
- one Property Value.

2.2.2.1.2.3 Extended Interface Object

4.3 Insert new type.

2.2.2.1.2.3.1 Definition

Extended Interface Objects shall in full comply with the common data structure as specified in Figure 12. Every Interface Object shall consist of a number of Properties. Every Property shall consist of

- one Property Description and
- one Property Value.

2.2.2.1.2.3.2 Property Description

The Property Description shall consist of all elements as specified in Full Interface Objects. Additionally it shall contain of

- the Datapoint Type

Datapoint Type (DPT)

The Datapoint Type shall describe the Datapoint Type according to which the Property is encoded. These Datapoint Type are specified in [07]. Every Extended Interface Object shall provide the correct DPT for its Interface Object Properties.

The value of the Datapoint Type shall be encoded in the fields *DPT (main number)* and *DPT (subnumber)* of the A_PropertyExtDescription_Response-PDU.

2.2.2.1.2.3.3 Property Value

The value of a Property shall be an array with array index 1 to max_nr_of_elem. The array element ´0´ shall contain the current number (unsigned16) of valid array elements. The array can be reset to no elements by writing zero on element ´0´. The array shall automatically be extended if an element is written beyond the currently last element, but within the maximum allowed number of entries.

2.2.2.1.2.4 Error handling

4.3.3 shall be appended with the following text.

If Extended Property Services are supported then Property arrays can be larger than 4095 entries. If the description of such an Interface Object is requested by using the service A_PropertyDescription_Read then the answer shall contain as max_nr_of_elem 4095.

2.3 Resource definition or used Resources

2.3.1 PID_IO_LIST

4.3.22 PID_IO_LIST shall be supplemented with the following text.

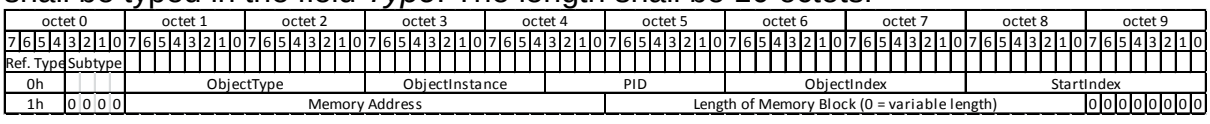
This Property is mandatory if the Extended Data Property Services and/or the Extended Function Property Services are implemented.

2.3.2 PDT_Reference

AN134 2.7.2 Change the Format of PDT_REFERENCE to the following:

Format

The format of the Property Value according PDT_REFERENCE shall be variable. It shall be typed in the field *Type*. The length shall be 10 octets.



AN134 2.7.2 Add to Reference Type 0 of PDT_REFERENCE following text:

The data in this Property does not indicate which methods are supported by the MaS for accessing Properties. See also 3.5.

2.4 Management Procedures

2.4.1 Device Management procedures

2.4.1.1 DM_InterfaceObjectWrite

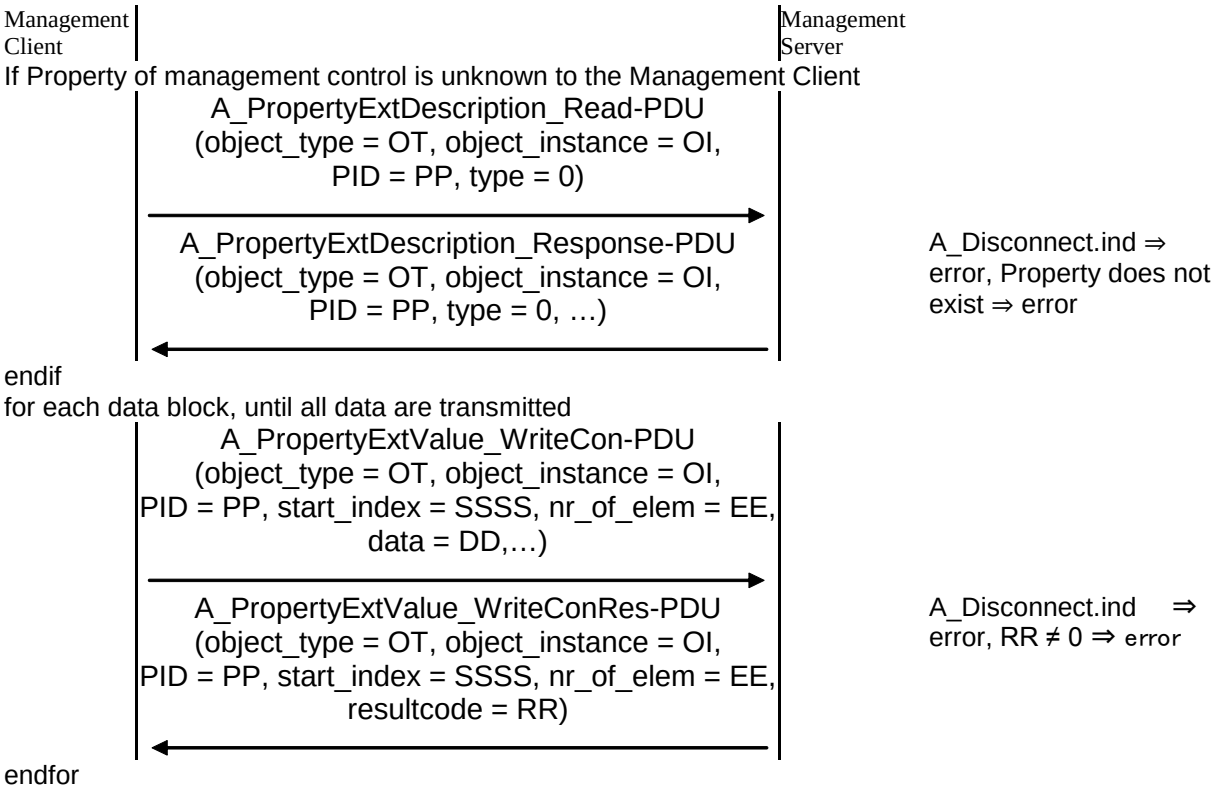
2.4.1.1.1 Procedure: DMP_ExtInterfaceObjectWriteCon_R

This Management Procedure shall use the connection oriented or connectionless communication mode.

Used Application Layer Services for Management

- A_PropertyExtDescription_Read
- A_PropertyExtValue_WriteCon

Sequence



Exception handling

The general exception handling shall apply.

The MaC shall not interpret the value of the Property Index contained in the A_Property_-Description_Response-PDU at the level of this Management Procedure. Possibly, error handling in case an unexpected value of the Property Index can be handled at the level of the Configuration Procedure in which this Management Procedure is used.

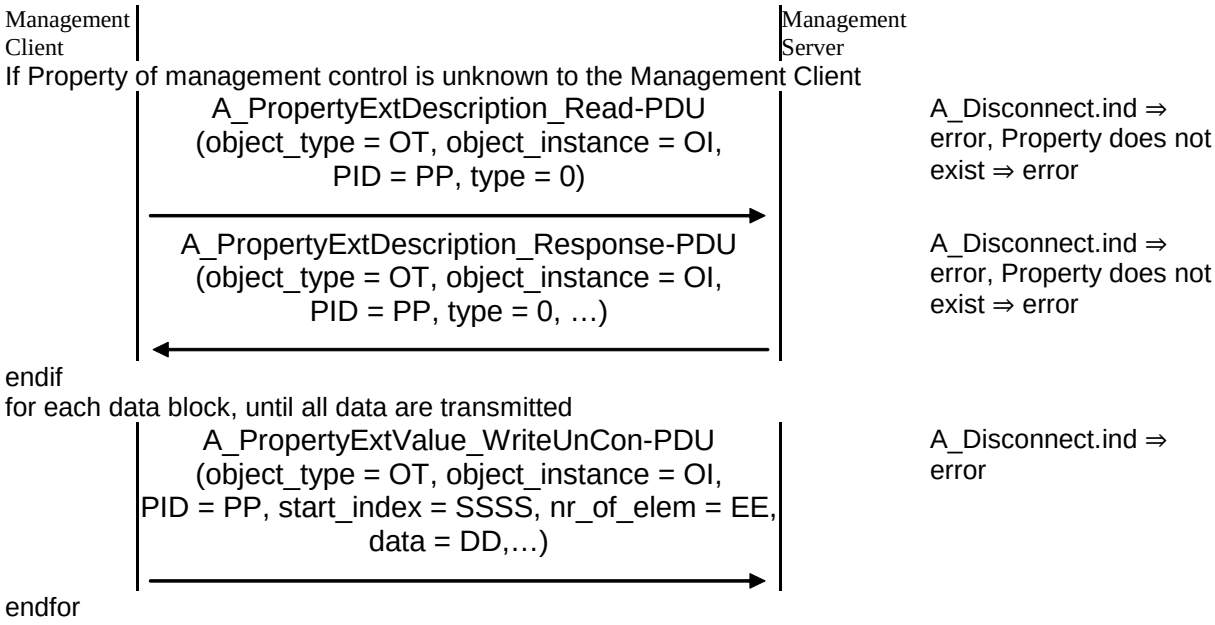
2.4.1.1.2 Procedure: DMP_ExtInterfaceObjectWriteUnCon_R

This Management Procedure shall use the connection oriented - or connectionless communication mode.

Used Application Layer Services for Management

- A_PropertyExtDescription_Read
- A_PropertyExtValue_WriteUnCon

Sequence



Exception handling

The general exception handling shall apply.

The MaC shall not interpret the value of the Property Index contained in the A_Property_-Description_Response-PDU at the level of this Management Procedure. Possibly, error handling in case an unexpected value of the Property Index can be handled at the level of the Configuration Procedure in which this Management Procedure is used.

2.4.1.2 DM_InterfaceObjectVerify

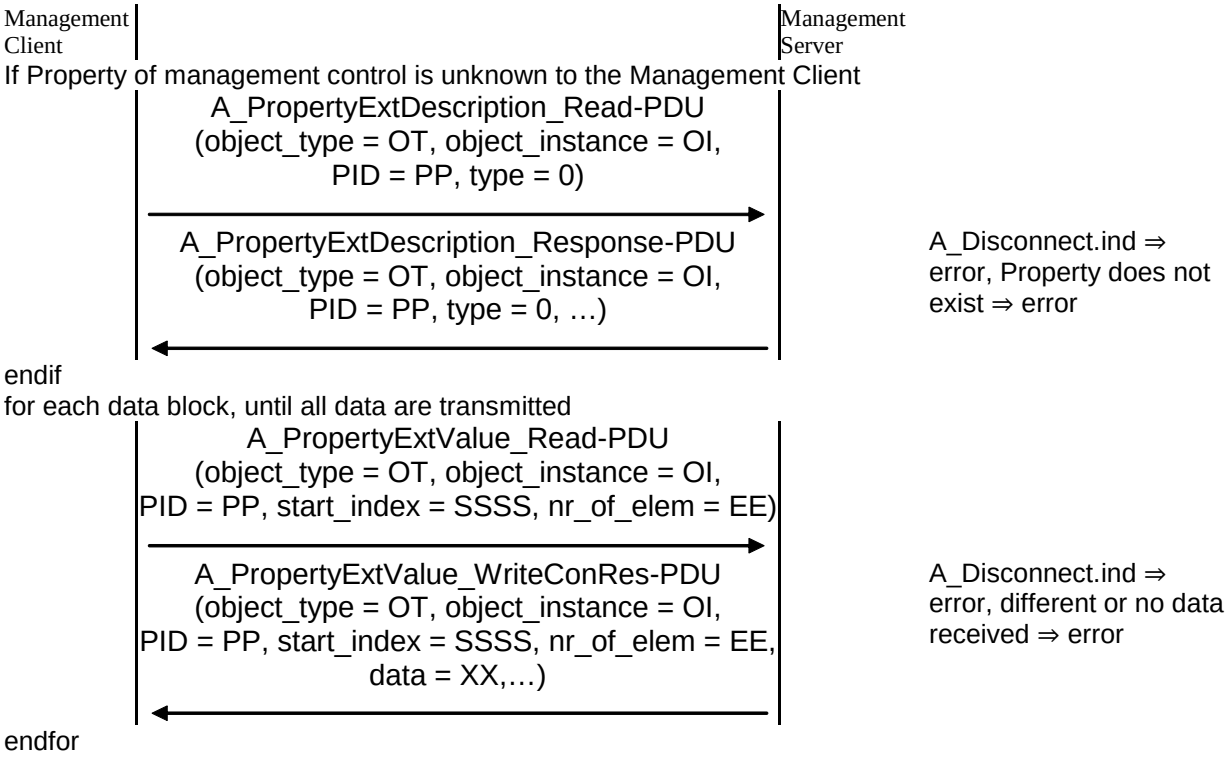
2.4.1.2.1 Procedure: DMP_ExtInterfaceObjectVerify_R

This Management Procedure shall use the connection oriented or connectionless communication mode.

Used Application Layer Services for Management

- A_PropertyExtDescription_Read
- A_PropertyExtValue_Read

Sequence



Exception handling

The general exception handling shall apply.

The Management Client shall not interpret the value of the Property Index contained in the A_Property_Description_Response-PDU at the level of this Management Procedure. Possibly, error handling in case an unexpected value of the Property Index can be handled at the level of the Configuration Procedure in which this Management Procedure is used.

2.4.1.3 DM_InterfaceObjectRead

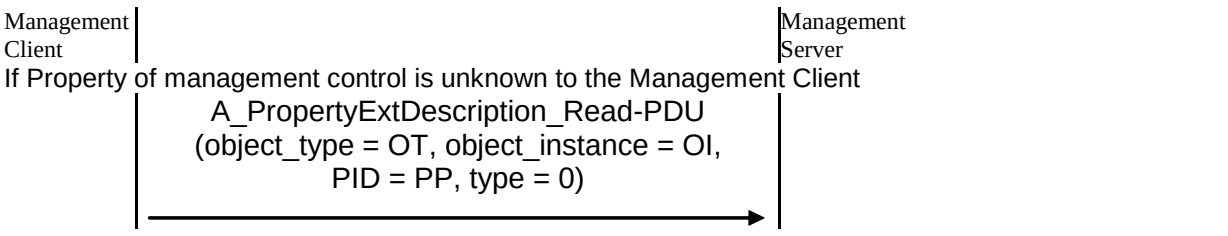
2.4.1.3.1 Procedure: DMP_ExtInterfaceObjectRead_R

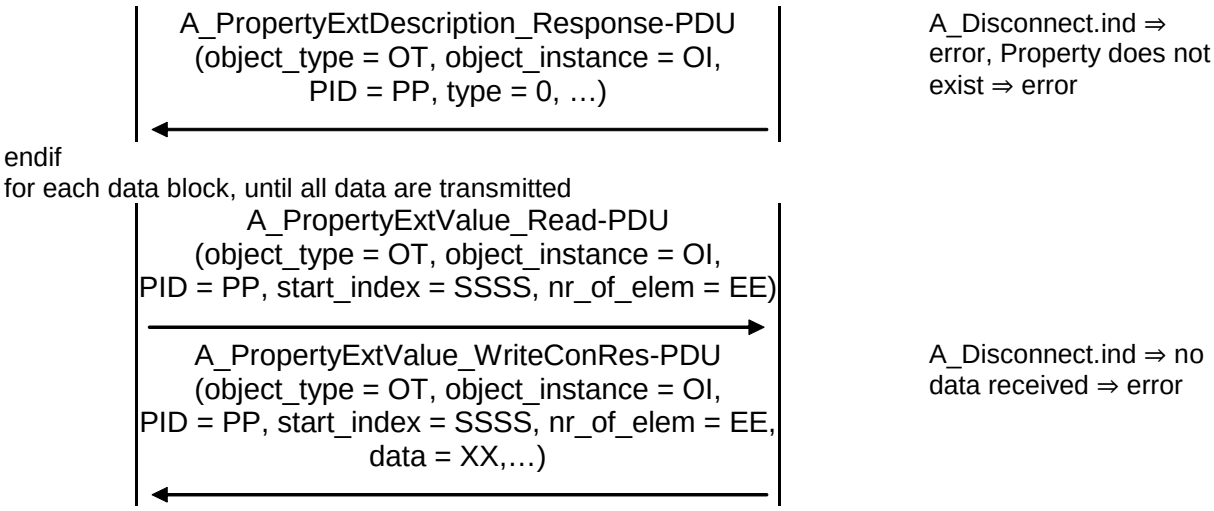
This Management Procedure shall use the connection oriented - or connectionless communication mode.

Used Application Layer Services for Management

- A_PropertyExtDescription_Read
- A_PropertyExtValue_Read

Sequence





Exception handling

The general exception handling shall apply.

The MaC shall not interpret the value of the Property Index contained in the A_Property_-Description_Response-PDU at the level of this Management Procedure. Possibly, error handling in case an unexpected value of the Property Index can be handled at the level of the Configuration Procedure in which this Management Procedure is used.

2.4.1.4 DM_InterfaceObjectScan

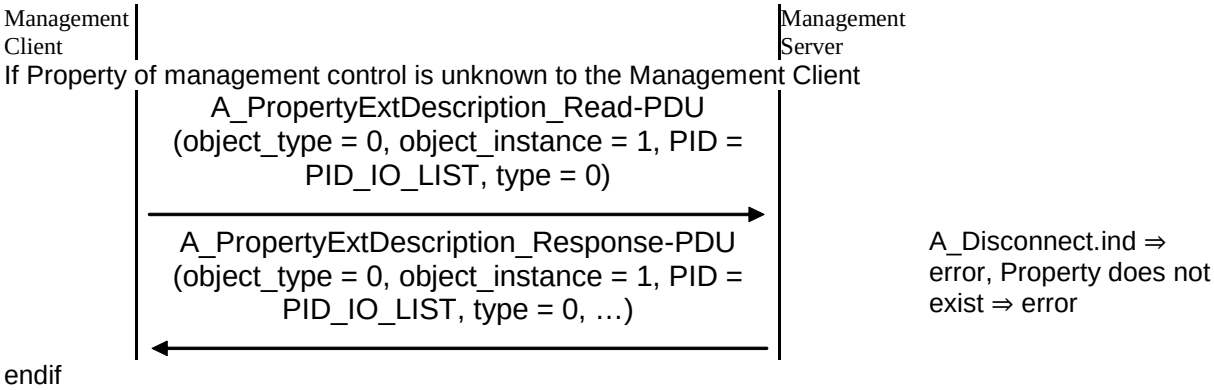
2.4.1.4.1 Procedure: DMP_ExtInterfaceObjectScan_R

This Management Procedure shall use the connection oriented - or connectionless communication mode.

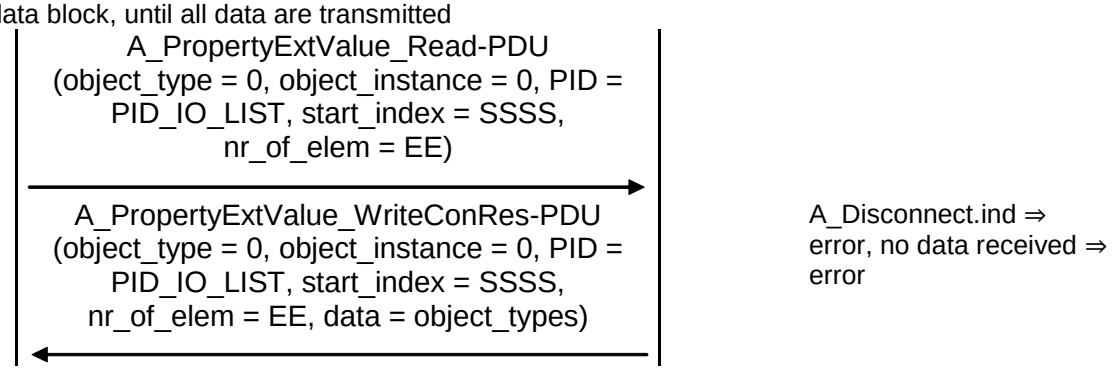
Used Application Layer Services for Management

- A_PropertyExtDescription_Read
- A_PropertyExtValue_Read

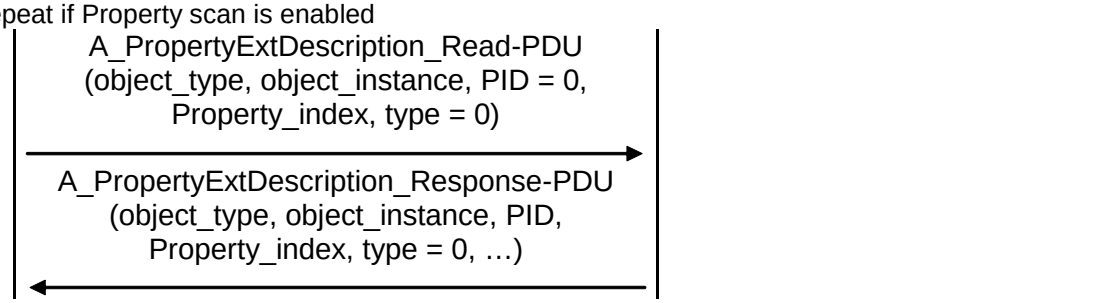
Sequence



io_list = new[]{}
for each data block, until all data are transmitted



io_list.Add(object_types)
endfor
Calculate Object Informations from io_list (object_index, object_type, object_instance)
foreach (object_index, object_type, object_instance) in io_list
Property_index = 0
repeat if Property scan is enabled



Property_index ++
Until PID = 0
endfor

Exception handling

The general exception handling shall apply.

2.4.1.5 DM_FunctionProperty_Write_R

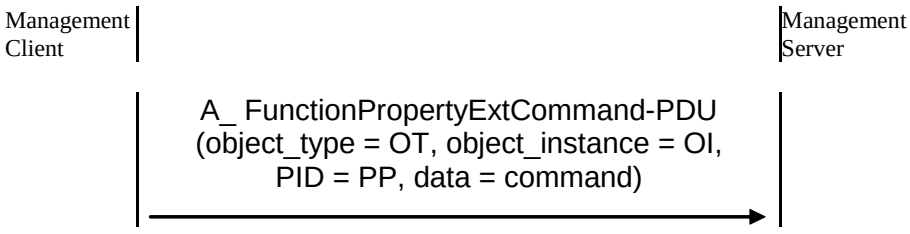
2.4.1.5.1 Procedure: DMP_ExtFunctionProperty_Write_R

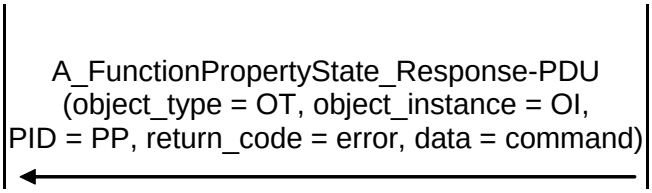
This Management Procedure shall use the connection oriented or connectionless communication mode.

Used Application Layer Services for Management

- A_FunctionPropertyExtCommand

Sequence





The Management Server shall execute the Function Property and return the result and error indication to the Management Client

Exception handling

The error shall be Function Property specific and is specified in [02]. The handling of this error depends on the Configuration Procedure in which this Management Procedure is used.

2.5 Configuration Procedures

This clause shall not be integrated in the KNX Specifications.

This version of this document does not introduce any Configuration Procedures.

2.6 Usage and context

See clause 1.

2.7 Profile definition

This Extended Interface Objects should be optional for all existing Profiles that support Properties.

2.8 Identifiers and discovery

This clause shall not be integrated in the KNX Specifications.

Currently there is a discussion [KSG00176] about identifying device features. This method shall be used then for identifying this new feature.

2.8.1 Property Identifiers Assignment scheme

Assignment Scheme in Chapter 3/7/3 3.1 shall be replaced by the following table.

Property Identifier	Object Type		
	Standardized System Interface Object Types (Object Types [0…99])	**Standardized Application Interface Object Types** (Object Types [100…50.000])	**Non-standardized Interface Object Types** (Object Types [50.001…65.535])
0 – 50			
51 – 154			
155 – 200			
201 – 255			
256 – 511			
512 – 3 071			
3 072 – 4 095			

NOTE 3 Property Identifiers over 255 cannot be used for Properties that should be accessible with A_GroupPropValue-Services.

3 Impact and dependencies

3.1 System specification (“Handbook”) dependencies

This clause is not intended for integration in the KNX Specifications.

Indications about the integration of the normative parts of this Application Note are given at the appropriate locations.

3.2 Configuration interworking

This clause is not intended for integration in the KNX Specifications.

This document does not contain any Configuration Procedures.

3.3 Runtime Interworking

This clause is not intended for integration in the KNX Specifications.

Not applicable.

3.4 Registration and certification

This clause is not intended for integration in the KNX Specifications.

The new feature described in this document is optional. It is allowed for new registrations.

3.5 Integration and common tool impact

This clause is not intended for integration in the KNX Specifications.

These services shall be integrated into ETS in order to write Parameters to Interface Objects with an index greater 255. If a device supports this feature shall be specified in the ETS database entry.

3.6 Risks and compatibility issues

This clause is not intended for integration in the KNX Specifications.

There are exist Interface Objects (Index > 255) that are only accessible by these new services.

There are exist Property Identifiers (> 255) that are only accessible by these new services.

With the new services the size of Property Arrays may be larger than 4095. These arrays cannot be read completely with the old services.

3.6.1 A_PropertyValue_Read

This service can only access Properties with Object Index ≤ 255 and PID ≤ 255. If a Property array has a size larger than 4 095 than only the first 4 095 entries can be read.

3.6.2 A_PropertyValue_Write

This service can only access Properties with Object Index ≤ 255 and PID ≤ 255. If a Property array has a size larger than 4 095 than only the first 4 095 entries can be written.

3.6.3 A_PropertyDescription_Read

Addressing with Property Identifier

This service can only access Properties with Object Index ≤ 255 and PID ≤ 255. If a Property array has a size bigger than 4095 then it shall replay 4095 as max_nr_of_elem.
(See 2.2.2.1.2.4.)

Addressing with Property Index

This service can only access Properties with Object Index ≤ 255 and Property Index ≤ 255. If a Property array has a size bigger than 4095 then it shall replay 4095 as max_nr_of_elem. If a Property is addressed with a Property Identifier >255 then the max_nr_of_elem of the A_PropertyDescription_Response-PDU shall be zero.

3.6.4 A_FunctionPropertyCommand

This service can only access Properties with Object Index ≤ 255 and PID ≤ 255. There are no further limitations.

3.6.5 A_FunctionPropertyState_Read

This service can only access Properties with Object Index ≤ 255 and PID ≤ 255. There are no further limitations.