9.5 Abstract syntax of COSEM PDUs

The abstract syntax of COSEM PDUs is specified in this subclause using ASN.1. See ISO/IEC 8824:2008.

NOTE The CIASE APDUs are specified in 10.4.9.

```
-- Preparation for Green Book 8.3
-- 2017-07-19
-- Prepared by Milan Kozole


COSEMpdu DEFINITIONS ::= BEGIN

ACSE-APDU ::= CHOICE
{
    aarq                            AARQ-apdu,
    aare                            AARE-apdu,
    rlrq                            RLRQ-apdu,          -- OPTIONAL
    rlre                            RLRE-apdu           -- OPTIONAL
}

XDLMS-APDU ::= CHOICE
{
-- standardised xDLMS pdus used in DLMS/COSEM

-- with no ciphering

    initiateRequest                 [1]  IMPLICIT       InitiateRequest,
    readRequest                     [5]  IMPLICIT       ReadRequest,
    writeRequest                    [6]  IMPLICIT       WriteRequest,

    initiateResponse                [8]  IMPLICIT       InitiateResponse,
    readResponse                    [12] IMPLICIT       ReadResponse,
    writeResponse                   [13] IMPLICIT       WriteResponse,

    confirmedServiceError           [14]                ConfirmedServiceError,

-- data-notification

    data-notification               [15] IMPLICIT       Data-Notification,

    unconfirmedWriteRequest         [22] IMPLICIT       UnconfirmedWriteRequest,
    informationReportRequest        [24] IMPLICIT       InformationReportRequest,

-- The APDU tag of each ciphered xDLMS APDU indicates the type of the unciphered APDU and whether
-- global or dedicated key is used. The type of the key is carried by the security header, and
-- after removing the encryption and/or verifying the authentication tag, the original APDU with
-- its APDU TAG is restored. Therefore, the APDU tags of the ciphered APDUs carry redundant
-- information, but they are retained for consistency.

-- with global ciphering

    glo-initiateRequest             [33]  IMPLICIT      OCTET STRING,
    glo-readRequest                 [37]  IMPLICIT      OCTET STRING,
    glo-writeRequest                [38]  IMPLICIT      OCTET STRING,

    glo-initiateResponse            [40]  IMPLICIT      OCTET STRING,
    glo-readResponse                [44]  IMPLICIT      OCTET STRING,
    glo-writeResponse               [45]  IMPLICIT      OCTET STRING,

    glo-confirmedServiceError       [46]  IMPLICIT      OCTET STRING,

    glo-unconfirmedWriteRequest     [54]  IMPLICIT      OCTET STRING,
    glo-informationReportRequest    [56]  IMPLICIT      OCTET STRING,

-- with dedicated ciphering

    ded-initiateRequest             [65]  IMPLICIT      OCTET STRING,
    ded-readRequest                 [69]  IMPLICIT      OCTET STRING,
    ded-writeRequest                [70]  IMPLICIT      OCTET STRING,

    ded-initiateResponse            [72]  IMPLICIT      OCTET STRING,
    ded-readResponse                [76]  IMPLICIT      OCTET STRING,
    ded-writeResponse               [77]  IMPLICIT      OCTET STRING,

    ded-confirmedServiceError       [78]  IMPLICIT      OCTET STRING,

    ded-unconfirmedWriteRequest     [86]  IMPLICIT      OCTET STRING,
    ded-informationReportRequest    [88]  IMPLICIT      OCTET STRING,

-- xDLMS APDUs used with LN referencing
```

```
-- with no ciphering

    get-request                     [192] IMPLICIT      Get-Request,
    set-request                     [193] IMPLICIT      Set-Request,
    event-notification-request      [194] IMPLICIT      EventNotificationRequest,
    action-request                  [195] IMPLICIT      Action-Request,

    get-response                    [196] IMPLICIT      Get-Response,
    set-response                    [197] IMPLICIT      Set-Response,
    action-response                 [199] IMPLICIT      Action-Response,

-- with global ciphering

    glo-get-request                 [200] IMPLICIT      OCTET STRING,
    glo-set-request                 [201] IMPLICIT      OCTET STRING,
    glo-event-notification-request  [202] IMPLICIT      OCTET STRING,
    glo-action-request              [203] IMPLICIT      OCTET STRING,

    glo-get-response                [204] IMPLICIT      OCTET STRING,
    glo-set-response                [205] IMPLICIT      OCTET STRING,
    glo-action-response             [207] IMPLICIT      OCTET STRING,

-- with dedicated ciphering

    ded-get-request                 [208] IMPLICIT      OCTET STRING,
    ded-set-request                 [209] IMPLICIT      OCTET STRING,
    ded-event-notification-request  [210] IMPLICIT      OCTET STRING,
    ded-actionRequest               [211] IMPLICIT      OCTET STRING,

    ded-get-response                [212] IMPLICIT      OCTET STRING,
    ded-set-response                [213] IMPLICIT      OCTET STRING,
    ded-action-response             [215] IMPLICIT      OCTET STRING,

-- the exception response pdu

    exception-response              [216] IMPLICIT      ExceptionResponse,


-- access

    access-request                  [217] IMPLICIT      Access-Request,
    access-response                 [218] IMPLICIT      Access-Response,

-- general APDUs
    general-glo-ciphering           [219] IMPLICIT      General-Glo-Ciphering,
    general-ded-ciphering           [220] IMPLICIT      General-Ded-Ciphering,
    general-ciphering               [221] IMPLICIT      General-Ciphering,
    general-signing                 [223] IMPLICIT      General-Signing,
    general-block-transfer          [224] IMPLICIT      General-Block-Transfer

-- The tags 230 and 231 are reserved for DLMS Gateway
-- reserved                        [230]
-- reserved                        [231]
}

AARQ-apdu ::= [APPLICATION 0] IMPLICIT SEQUENCE
{
-- [APPLICATION 0] == [ 60H ] = [ 96 ]

    protocol-version                [0] IMPLICIT        BIT STRING {version1 (0)}
                                                        DEFAULT{version1},
    application-context-name        [1]                 Application-context-name,
    called-AP-title                 [2]                 AP-title OPTIONAL,
    called-AE-qualifier             [3]                 AE-qualifier OPTIONAL,
    called-AP-invocation-id         [4]                 AP-invocation-identifier OPTIONAL,
    called-AE-invocation-id         [5]                 AE-invocation-identifier OPTIONAL,
    calling-AP-title                [6]                 AP-title OPTIONAL,
    calling-AE-qualifier            [7]                 AE-qualifier OPTIONAL,
    calling-AP-invocation-id        [8]                 AP-invocation-identifier OPTIONAL,
    calling-AE-invocation-id        [9]                 AE-invocation-identifier OPTIONAL,

-- The following field shall not be present if only the kernel is used.
    sender-acse-requirements        [10] IMPLICIT      ACSE-requirements OPTIONAL,

-- The following field shall only be present if the authentication functional unit is selected.
    mechanism-name                  [11] IMPLICIT      Mechanism-name OPTIONAL,

-- The following field shall only be present if the authentication functional unit is selected.

    calling-authentication-value    [12] EXPLICIT      Authentication-value OPTIONAL,
    implementation-information      [29] IMPLICIT      Implementation-data OPTIONAL,
    user-information                [30] EXPLICIT      Association-information OPTIONAL
}
```

```
-- The user-information field shall carry an InitiateRequest APDU encoded in A-XDR, and then
-- encoding the resulting OCTET STRING in BER.

AARE-apdu ::= [APPLICATION 1] IMPLICIT SEQUENCE
{
-- [APPLICATION 1] == [ 61H ] = [ 97 ]

    protocol-version                    [0] IMPLICIT        BIT STRING {version1 (0)}
                                                            DEFAULT {version1},
    application-context-name            [1]                 Application-context-name,
    result                              [2]                 Association-result,
    result-source-diagnostic            [3]                 Associate-source-diagnostic,
    responding-AP-title                 [4]                 AP-title OPTIONAL,
    responding-AE-qualifier             [5]                 AE-qualifier OPTIONAL,
    responding-AP-invocation-id         [6]                 AP-invocation-identifier OPTIONAL,
    responding-AE-invocation-id         [7]                 AE-invocation-identifier OPTIONAL,

-- The following field shall not be present if only the kernel is used.
    responder-acse-requirements         [8] IMPLICIT        ACSE-requirements OPTIONAL,

-- The following field shall only be present if the authentication functional unit is selected.
    mechanism-name                      [9] IMPLICIT        Mechanism-name OPTIONAL,

-- The following field shall only be present if the authentication functional unit is selected.
    responding-authentication-value     [10] EXPLICIT       Authentication-value OPTIONAL,
    implementation-information          [29] IMPLICIT       Implementation-data OPTIONAL,
    user-information                    [30] EXPLICIT       Association-information OPTIONAL
}

-- The user-information field shall carry either an InitiateResponse (or, when the proposed xDLMS
-- context is not accepted by the server, a ConfirmedServiceError) APDU encoded in A-XDR, and then
-- encoding the resulting OCTET STRING in BER.

RLRQ-apdu ::= [APPLICATION 2] IMPLICIT SEQUENCE
{
-- [APPLICATION 2] == [ 62H ] = [ 98 ]

    reason                              [0] IMPLICIT        Release-request-reason OPTIONAL,
    user-information                    [30] EXPLICIT       Association-information OPTIONAL
}

RLRE-apdu ::= [APPLICATION 3] IMPLICIT SEQUENCE
{
-- [APPLICATION 3] == [ 63H ] = [ 99 ]

    reason                              [0] IMPLICIT        Release-response-reason OPTIONAL,
    user-information                    [30] EXPLICIT       Association-information OPTIONAL
}

-- The user-information field of the RLRQ / RLRE APDU may carry an InitiateRequest APDU encoded in
-- A-XDR, and then encoding the resulting OCTET STRING in BER, when the AA to be released uses
-- ciphering.

-- types used in the fields of the ACSE APDUs, in the order of their occurrence

Application-context-name ::=        OBJECT IDENTIFIER

AP-title ::=                        OCTET STRING

AE-qualifier ::=                    OCTET STRING

AP-invocation-identifier ::=        INTEGER

AE-invocation-identifier ::=        INTEGER

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

Mechanism-name ::=                  OBJECT IDENTIFIER

Authentication-value ::= CHOICE
{
    charstring                      [0] IMPLICIT    GraphicString,
    bitstring                       [1] IMPLICIT    BIT STRING
}

Implementation-data ::=             GraphicString

Association-information ::=         OCTET STRING

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

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

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

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


-- Useful types

Integer8 ::=                          INTEGER(-128..127)
Integer16 ::=                         INTEGER(-32768..32767)
Integer32 ::=                         INTEGER(-2147483648..2147483647)
Integer64 ::=                         INTEGER(-9223372036854775808..9223372036854775807)
Unsigned8 ::=                         INTEGER(0..255)
Unsigned16 ::=                        INTEGER(0..65535)
Unsigned32 ::=                        INTEGER(0..4294967295)
Unsigned64 ::=                        INTEGER(0..18446744073709551615)


-- xDLMS APDU-s used during Association establishment

InitiateRequest ::= SEQUENCE
{
--  shall not be encoded in DLMS without ciphering
    dedicated-key                     OCTET STRING OPTIONAL,
    response-allowed                  BOOLEAN DEFAULT TRUE,
    proposed-quality-of-service       [0] IMPLICIT Integer8 OPTIONAL,
    proposed-dlms-version-number      Unsigned8,
    proposed-conformance              Conformance, -- Shall be encoded in BER
    client-max-receive-pdu-size       Unsigned16
}

-- In DLMS/COSEM, the quality-of-service parameter is not used. Any value shall be accepted.

-- The Conformance field shall be encoded in BER. See IEC 61334-6 Example 1.

InitiateResponse ::= SEQUENCE
{
    negotiated-quality-of-service     [0] IMPLICIT Integer8 OPTIONAL,
    negotiated-dlms-version-number    Unsigned8,
    negotiated-conformance            Conformance, -- Shall be encoded in BER
    server-max-receive-pdu-size       Unsigned16,
    vaa-name                          ObjectName
}

-- In the case of LN referencing, the value of the vaa-name is 0x0007
-- In the case of SN referencing, the value of the vaa-name is the base name of the
-- Current Association object, 0xFA00
-- Conformance Block
```

```
-- SIZE constrained BIT STRING is extension of ASN.1 notation

Conformance ::= [APPLICATION 31] IMPLICIT BIT STRING
{
    -- the bit is set when the corresponding service or functionality is available
    reserved-zero                       (0),
    -- The actual list of general protection services depends on the security suite
    general-protection                  (1),
    general-block-transfer              (2),
    read                                (3),
    write                               (4),
    unconfirmed-write                   (5),
    reserved-six                        (6),
    reserved-seven                      (7),
    attribute0-supported-with-set       (8),
    priority-mgmt-supported             (9),
    attribute0-supported-with-get       (10),
    block-transfer-with-get-or-read     (11),
    block-transfer-with-set-or-write    (12),
    block-transfer-with-action          (13),
    multiple-references                 (14),
    information-report                  (15),
    data-notification                   (16),
    access                              (17),
    parameterized-access                (18),
    get                                 (19),
    set                                 (20),
    selective-access                    (21),
    event-notification                  (22),
    action                              (23)
}

ObjectName ::=                              Integer16
-- for named variable objects (short names), the last three bits shall be set to 000;
-- for vaa-name objects, the last three bits shall be set to 111.

-- The Confirmed ServiceError APDU is used only with the InitiateRequest, ReadRequest and
-- WriteRequest APDUs when the request fails, to provide diagnostic information.

ConfirmedServiceError ::= CHOICE
{
-- tag 0 is reserved
-- In DLMS/COSEM only initiateEror, read and write are relevant

    initiateError                       [1] ServiceError,
    getStatus                           [2] ServiceError,
    getNameList                         [3] ServiceError,
    getVariableAttribute                [4] ServiceError,
    read                                [5] ServiceError,
    write                               [6] ServiceError,
    getDataSetAttribute                 [7] ServiceError,
    getTIAttribute                      [8] ServiceError,
    changeScope                         [9] ServiceError,
    start                               [10] ServiceError,
    stop                                [11] ServiceError,
    resume                              [12] ServiceError,
    makeUsable                          [13] ServiceError,
    initiateLoad                        [14] ServiceError,
    loadSegment                         [15] ServiceError,
    terminateLoad                       [16] ServiceError,
    initiateUpLoad                      [17] ServiceError,
    upLoadSegment                       [18] ServiceError,
    terminateUpLoad                     [19] ServiceError
}

ServiceError  ::= CHOICE
{
    application-reference               [0] IMPLICIT ENUMERATED
    {
    -- DLMS provider only
        other                               (0),
        time-elapsed                        (1),  -- time out since request sent
        application-unreachable             (2),  -- peer AEi not reachable
        application-reference-invalid       (3),  -- addressing trouble
        application-context-unsupported     (4),  -- application-context incompatibility
        provider-communication-error        (5),  -- error at the local or distant equipment
        deciphering-error                   (6)   -- error detected by the deciphering function
    },

    hardware-resource                   [1] IMPLICIT ENUMERATED
    {
    -- VDE hardware troubles
        other                               (0),
        memory-unavailable                  (1),
        processor-resource-unavailable      (2),
        mass-storage-unavailable            (3),
```

```
        other-resource-unavailable          (4)
    },

    vde-state-error                     [2] IMPLICIT ENUMERATED
    {
    -- Error source description
        other                               (0),
        no-dlms-context                     (1),
        loading-data-set                    (2),
        status-nochange                     (3),
        status-inoperable                   (4)
    },

    service                             [3] IMPLICIT ENUMERATED
    {
    -- service handling troubles
        other                               (0),
        pdu-size                            (1),  -- pdu too long
        service-unsupported                 (2)   -- as defined in the conformance block
    },

    definition                          [4] IMPLICIT ENUMERATED
    {
    -- object bound troubles in a service
        other                               (0),
        object-undefined                    (1),  -- object not defined at the VDE
        object-class-inconsistent           (2),  -- class of object incompatible with asked
                                                     service
        object-attribute-inconsistent       (3)   -- object attributes are inconsistent
    },

    access                              [5] IMPLICIT ENUMERATED
    {
    -- object access error
        other                               (0),
        scope-of-access-violated            (1),  -- access denied through authorisation reason
        object-access-violated              (2),  -- access incompatible with object attribute
        hardware-fault                      (3),  -- access fail for hardware reason
        object-unavailable                  (4)   -- VDE hands object for unavailable
    },

    initiate                            [6] IMPLICIT ENUMERATED
    {
    -- initiate service error
        other                               (0),
        dlms-version-too-low                (1),  -- proposed DLMS version too low
        incompatible-conformance            (2),  -- proposed service not sufficient
        pdu-size-too-short                  (3),  -- proposed PDU size too short
        refused-by-the-VDE-Handler          (4)   -- vaa creation impossible or not allowed
    },

    load-data-set                       [7] IMPLICIT ENUMERATED
    {
    -- data set load services error
        other                               (0),
        primitive-out-of-sequence           (1),  -- according to the DataSet loading state
                                                     transitions
        not-loadable                        (2),  -- loadable attribute set to FALSE
        dataset-size-too-large              (3),  -- evaluated Data Set size too large
        not-awaited-segment                 (4),  -- proposed segment not awaited
        interpretation-failure              (5),  -- segment interpretation error
        storage-failure                     (6),  -- segment storage error
        data-set-not-ready                  (7)   -- Data Set not in correct state for uploading
    },

    -- change-scope                     [8] IMPLICIT ENUMERATED

    task                                [9] IMPLICIT ENUMERATED
    {
    -- TI services error
        other                               (0),
        no-remote-control                   (1),  -- Remote Control parameter set to FALSE
        ti-stopped                          (2),  -- TI in stopped state
        ti-running                          (3),  -- TI in running state
        ti-unusable                         (4)   -- TI in unusable state
    }

    -- other                            [10] IMPLICIT ENUMERATED
}
```

```
-- COSEM APDUs using short name referencing

ReadRequest ::= SEQUENCE OF Variable-Access-Specification

ReadResponse ::= SEQUENCE OF CHOICE
{
    data                                [0] Data,
    data-access-error                   [1] IMPLICIT Data-Access-Result,
    data-block-result                   [2] IMPLICIT Data-Block-Result,
    block-number                        [3] IMPLICIT Unsigned16
}

WriteRequest  ::= SEQUENCE
{
    variable-access-specification       SEQUENCE OF Variable-Access-Specification,
    list-of-data                        SEQUENCE OF Data
}

WriteResponse ::= SEQUENCE OF CHOICE
{
    success                             [0] IMPLICIT NULL,
    data-access-error                   [1] IMPLICIT Data-Access-Result,
    block-number                        [2] Unsigned16
}

UnconfirmedWriteRequest ::= SEQUENCE
{
    variable-access-specification       SEQUENCE OF Variable-Access-Specification,
    list-of-data                        SEQUENCE OF Data
}

InformationReportRequest ::= SEQUENCE
{
    current-time                        GeneralizedTime OPTIONAL,
    variable-access-specification       SEQUENCE OF Variable-Access-Specification,
    list-of-data                        SEQUENCE OF Data
}


-- COSEM APDUs using logical name referencing

Get-Request ::= CHOICE
{
    get-request-normal                  [1] IMPLICIT    Get-Request-Normal,
    get-request-next                    [2] IMPLICIT    Get-Request-Next,
    get-request-with-list               [3] IMPLICIT    Get-Request-With-List
}

Get-Request-Normal ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    cosem-attribute-descriptor          Cosem-Attribute-Descriptor,
    access-selection                    Selective-Access-Descriptor OPTIONAL
}

Get-Request-Next ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    block-number                        Unsigned32
}

Get-Request-With-List ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    attribute-descriptor-list           SEQUENCE OF Cosem-Attribute-Descriptor-With-Selection
}

Get-Response ::= CHOICE
{
    get-response-normal                 [1] IMPLICIT    Get-Response-Normal,
    get-response-with-datablock         [2] IMPLICIT    Get-Response-With-Datablock,
    get-response-with-list              [3] IMPLICIT    Get-Response-With-List
}

Get-Response-Normal ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    result                              Get-Data-Result
}
Get-Response-With-Datablock ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    result                              DataBlock-G
}

Get-Response-With-List ::= SEQUENCE
```

```
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    result                             SEQUENCE OF Get-Data-Result
}

Set-Request ::= CHOICE
{
  set-request-normal                         [1] IMPLICIT  Set-Request-Normal,
  set-request-with-first-datablock           [2] IMPLICIT  Set-Request-With-First-Datablock,
  set-request-with-datablock                 [3] IMPLICIT  Set-Request-With-Datablock,
  set-request-with-list                      [4] IMPLICIT  Set-Request-With-List,
  set-request-with-list-and-first-datablock  [5] IMPLICIT  Set-Request-With-List-And-First-Datablock
}

Set-Request-Normal ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    cosem-attribute-descriptor         Cosem-Attribute-Descriptor,
    access-selection                   Selective-Access-Descriptor OPTIONAL,
    value                              Data
}

Set-Request-With-First-Datablock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    cosem-attribute-descriptor         Cosem-Attribute-Descriptor,
    access-selection                   [0] IMPLICIT Selective-Access-Descriptor OPTIONAL,
    datablock                          DataBlock-SA
}

Set-Request-With-Datablock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    datablock                          DataBlock-SA
}

Set-Request-With-List ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    attribute-descriptor-list          SEQUENCE OF Cosem-Attribute-Descriptor-With-Selection,
    value-list                         SEQUENCE OF Data
}

Set-Request-With-List-And-First-Datablock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    attribute-descriptor-list          SEQUENCE OF Cosem-Attribute-Descriptor-With-Selection,
    datablock                          DataBlock-SA
}

Set-Response ::= CHOICE
{
    set-response-normal                    [1] IMPLICIT   Set-Response-Normal,
    set-response-datablock                 [2] IMPLICIT   Set-Response-Datablock,
    set-response-last-datablock            [3] IMPLICIT   Set-Response-Last-Datablock,
    set-response-last-datablock-with-list  [4] IMPLICIT   Set-Response-Last-Datablock-With-List,
    set-response-with-list                 [5] IMPLICIT   Set-Response-With-List
}

Set-Response-Normal ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    result                             Data-Access-Result
}

Set-Response-Datablock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    block-number                       Unsigned32
}

Set-Response-Last-Datablock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    result                             Data-Access-Result,
    block-number                       Unsigned32
}

Set-Response-Last-Datablock-With-List ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    result                             SEQUENCE OF Data-Access-Result,
    block-number                       Unsigned32
}

Set-Response-With-List ::= SEQUENCE
```

```
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    result                             SEQUENCE OF Data-Access-Result
}

Action-Request ::= CHOICE
{
  action-request-normal                      [1] IMPLICIT Action-Request-Normal,
  action-request-next-pblock                 [2] IMPLICIT Action-Request-Next-Pblock,
  action-request-with-list                   [3] IMPLICIT Action-Request-With-List,
  action-request-with-first-pblock           [4] IMPLICIT Action-Request-With-First-Pblock,
  action-request-with-list-and-first-pblock  [5] IMPLICIT Action-Request-With-List-And-First-Pblock,
  action-request-with-pblock                 [6] IMPLICIT Action-Request-With-Pblock
}

Action-Request-Normal ::= SEQUENCE
{
    invoke-id-and-priority              Invoke-Id-And-Priority,
    cosem-method-descriptor             Cosem-Method-Descriptor,
    method-invocation-parameters        Data OPTIONAL
}

Action-Request-Next-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    block-number                       Unsigned32
}

Action-Request-With-List ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    cosem-method-descriptor-list       SEQUENCE OF Cosem-Method-Descriptor,
    method-invocation-parameters       SEQUENCE OF Data
}

Action-Request-With-First-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    cosem-method-descriptor            Cosem-Method-Descriptor,
    pblock                             DataBlock-SA
}

Action-Request-With-List-And-First-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    cosem-method-descriptor-list       SEQUENCE OF Cosem-Method-Descriptor,
    pblock                             DataBlock-SA
}

Action-Request-With-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    pblock                             DataBlock-SA
}

Action-Response ::= CHOICE
{
    action-response-normal             [1] IMPLICIT    Action-Response-Normal,
    action-response-with-pblock        [2] IMPLICIT    Action-Response-With-Pblock,
    action-response-with-list          [3] IMPLICIT    Action-Response-With-List,
    action-response-next-pblock        [4] IMPLICIT    Action-Response-Next-Pblock
}

Action-Response-Normal ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    single-response                    Action-Response-With-Optional-Data
}

Action-Response-With-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    pblock                             DataBlock-SA
}

Action-Response-With-List ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    list-of-responses                  SEQUENCE OF Action-Response-With-Optional-Data
}

Action-Response-Next-Pblock ::= SEQUENCE
{
    invoke-id-and-priority             Invoke-Id-And-Priority,
    block-number                       Unsigned32
}
```

```
EventNotificationRequest ::= SEQUENCE
{
    time                                OCTET STRING  OPTIONAL,
    cosem-attribute-descriptor          Cosem-Attribute-Descriptor,
    attribute-value                     Data
}

ExceptionResponse ::= SEQUENCE
{
    state-error                         [0] IMPLICIT ENUMERATED
    {
        service-not-allowed                 (1),
        service-unknown                     (2)
    },
    service-error                       [1] CHOICE
    {
        operation-not-possible              [1] IMPLICIT NULL,
        service-not-supported               [2] IMPLICIT NULL,
        other-reason                        [3] IMPLICIT NULL,
        pdu-too-long                        [4] IMPLICIT NULL,
        deciphering-error                   [5] IMPLICIT NULL,
        invocation-counter-error            [6] IMPLICIT Unsigned32
    }
}


--       Access

Access-Request ::= SEQUENCE
{
   long-invoke-id-and-priority          Long-Invoke-Id-And-Priority,
   date-time                            OCTET STRING,
   access-request-body                  Access-Request-Body
}

Access-Response ::= SEQUENCE
{
        long-invoke-id-and-priority         Long-Invoke-Id-And-Priority,
        date-time                           OCTET STRING,
        access-response-body                Access-Response-Body
}


--       Data-Notification

Data-Notification ::= SEQUENCE
{
   long-invoke-id-and-priority          Long-Invoke-Id-And-Priority,
   date-time                            OCTET STRING,
   notification-body                    Notification-Body
}


-- General APDUs

General-Ded-Ciphering ::= SEQUENCE
{
   system-title                         OCTET STRING,
   ciphered-content                     OCTET STRING
}

General-Glo-Ciphering ::= SEQUENCE
{
   system-title                         OCTET STRING,
   ciphered-content                     OCTET STRING
}

General-Ciphering ::= SEQUENCE
{
   transaction-id                       OCTET STRING,
   originator-system-title              OCTET STRING,
   recipient-system-title               OCTET STRING,
   date-time                            OCTET STRING,
   other-information                    OCTET STRING,
   key-info                             Key-Info OPTIONAL,
   ciphered-content                     OCTET STRING
}
```

```
General-Signing ::= SEQUENCE
{
   transaction-id                       OCTET STRING,
   originator-system-title              OCTET STRING,
   recipient-system-title               OCTET STRING,
   date-time                            OCTET STRING,
   other-information                    OCTET STRING,
   content                              OCTET STRING,
   signature                            OCTET STRING
}

General-Block-Transfer ::= SEQUENCE
{
   block-control                        Block-Control,
   block-number                         Unsigned16,
   block-number-ack                     Unsigned16,
   block-data                           OCTET STRING
}


-- Types used in the xDLMS data transfer services

Variable-Access-Specification ::= CHOICE
{
    variable-name                       [2] IMPLICIT ObjectName,
-- detailed-access [3] is not used in DLMS/COSEM
    parameterized-access                [4] IMPLICIT Parameterized-Access,
    block-number-access                 [5] IMPLICIT Block-Number-Access,
    read-data-block-access              [6] IMPLICIT Read-Data-Block-Access,
    write-data-block-access             [7] IMPLICIT Write-Data-Block-Access
}

Parameterized-Access ::= SEQUENCE
{
    variable-name                       ObjectName,
    selector                            Unsigned8,
    parameter                           Data
}

Block-Number-Access ::= SEQUENCE
{
    block-number                        Unsigned16
}

Read-Data-Block-Access ::= SEQUENCE
{
    last-block                          BOOLEAN,
    block-number                        Unsigned16,
    raw-data                            OCTET STRING
}


Write-Data-Block-Access ::= SEQUENCE
{
   last-block                           BOOLEAN,
   block-number                         Unsigned16
}

Data ::= CHOICE
{
    null-data                           [0]   IMPLICIT   NULL,
    array                               [1]   IMPLICIT   SEQUENCE OF Data,
    structure                           [2]   IMPLICIT   SEQUENCE OF Data,
    boolean                             [3]   IMPLICIT   BOOLEAN,
    bit-string                          [4]   IMPLICIT   BIT STRING,
    double-long                         [5]   IMPLICIT   Integer32,
    double-long-unsigned                [6]   IMPLICIT   Unsigned32,
    octet-string                        [9]   IMPLICIT   OCTET STRING,
    visible-string                      [10]  IMPLICIT   VisibleString,
    utf8-string                         [12]  IMPLICIT   UTF8String,
    bcd                                 [13]  IMPLICIT   Integer8,
    integer                             [15]  IMPLICIT   Integer8,
    long                                [16]  IMPLICIT   Integer16,
    unsigned                            [17]  IMPLICIT   Unsigned8,
    long-unsigned                       [18]  IMPLICIT   Unsigned16,
    compact-array                       [19]  IMPLICIT   SEQUENCE
    {
        contents-description                [0]              TypeDescription,
        array-contents                      [1]   IMPLICIT   OCTET STRING
    },
    long64                              [20]  IMPLICIT   Integer64,
    long64-unsigned                     [21]  IMPLICIT   Unsigned64,
    enum                                [22]  IMPLICIT   Unsigned8,
    float32                             [23]  IMPLICIT   OCTET STRING (SIZE(4)),
    float64                             [24]  IMPLICIT   OCTET STRING (SIZE(8)),
    date-time                           [25]  IMPLICIT   OCTET STRING (SIZE(12)),
```

```
    date                                [26]  IMPLICIT   OCTET STRING (SIZE(5)),
    time                                [27]  IMPLICIT   OCTET STRING (SIZE(4)),
    dont-care                           [255] IMPLICIT   NULL
}

-- The following TypeDescription relates to the compact-array data Type

TypeDescription ::= CHOICE
{
    null-data                           [0]   IMPLICIT  NULL,
    array                               [1]   IMPLICIT  SEQUENCE
    {
        number-of-elements        Unsigned16,
        type-description          TypeDescription
    },
    structure                           [2]   IMPLICIT  SEQUENCE OF TypeDescription,
    boolean                             [3]   IMPLICIT  NULL,
    bit-string                          [4]   IMPLICIT  NULL,
    double-long                         [5]   IMPLICIT  NULL,
    double-long-unsigned                [6]   IMPLICIT  NULL,
    octet-string                        [9]   IMPLICIT  NULL,
    visible-string                      [10]  IMPLICIT  NULL,
    utf8-string                         [12]  IMPLICIT  NULL,
    bcd                                 [13]  IMPLICIT  NULL,
    integer                             [15]  IMPLICIT  NULL,
    long                                [16]  IMPLICIT  NULL,
    unsigned                            [17]  IMPLICIT  NULL,
    long-unsigned                       [18]  IMPLICIT  NULL,
    long64                              [20]  IMPLICIT  NULL,
    long64-unsigned                     [21]  IMPLICIT  NULL,
    enum                                [22]  IMPLICIT  NULL,
    float32                             [23]  IMPLICIT  NULL,
    float64                             [24]  IMPLICIT  NULL,
    date-time                           [25]  IMPLICIT  NULL,
    date                                [26]  IMPLICIT  NULL,
    time                                [27]  IMPLICIT  NULL,
    dont-care                           [255] IMPLICIT  NULL
}

Data-Access-Result ::= ENUMERATED
{
    success                             (0),
    hardware-fault                      (1),
    temporary-failure                   (2),
    read-write-denied                   (3),
    object-undefined                    (4),
    object-class-inconsistent           (9),
    object-unavailable                  (11),
    type-unmatched                      (12),
    scope-of-access-violated            (13),
    data-block-unavailable              (14),
    long-get-aborted                    (15),
    no-long-get-in-progress             (16),
    long-set-aborted                    (17),
    no-long-set-in-progress             (18),
    data-block-number-invalid           (19),
    other-reason                        (250)
}

Action-Result ::= ENUMERATED
{
    success                             (0),
    hardware-fault                      (1),
    temporary-failure                   (2),
    read-write-denied                   (3),
    object-undefined                    (4),
    object-class-inconsistent           (9),
    object-unavailable                  (11),
    type-unmatched                      (12),
    scope-of-access-violated            (13),
    data-block-unavailable              (14),
    long-action-aborted                 (15),
    no-long-action-in-progress          (16),
    other-reason                        (250)
}

-- IEC 61334-6 clause 5 specifies that bits of any byte are numbered from 1 to 8,
-- where bit 8 is the most significant.
-- In the DLMS UA Green Book, bits are numbered from 0 to 7.
-- Use of Invoke-Id-And-Priority
--    invoke-id               bits 0-3
--    reserved                bits 4-5
--    service-class           bit  6        0 = Unconfirmed, 1 = Confirmed
--    priority                bit  7        0 = Normal, 1 = High
Invoke-Id-And-Priority ::=            Unsigned8
```

```
-- Use of Long-Invoke-Id-And-Priority
--    long-invoke-id           bits 0-23
--    reserved                 bits 24-27
--    self-descriptive         bit  28       0 = Not-Self-Descriptive, 1 = Self-Descriptive
--    processing-option        bit  29       0 = Continue on Error, 1 = Break on Error
--    service-class            bit  30       0 = Unconfirmed, 1 = Confirmed
--    priority                 bit  31       0 = Normal, 1 = High
Long-Invoke-Id-And-Priority ::=        Unsigned32

Cosem-Attribute-Descriptor ::= SEQUENCE
{
    class-id                           Cosem-Class-Id,
    instance-id                        Cosem-Object-Instance-Id,
    attribute-id                       Cosem-Object-Attribute-Id
}

Cosem-Method-Descriptor ::= SEQUENCE
{
    class-id                           Cosem-Class-Id,
    instance-id                        Cosem-Object-Instance-Id,
    method-id                          Cosem-Object-Method-Id
}

Cosem-Class-Id ::=                     Unsigned16

Cosem-Object-Instance-Id ::=           OCTET STRING (SIZE(6))

Cosem-Object-Attribute-Id ::=          Integer8

Cosem-Object-Method-Id ::=             Integer8

Selective-Access-Descriptor ::= SEQUENCE
{
    access-selector                    Unsigned8,
    access-parameters                  Data
}

Cosem-Attribute-Descriptor-With-Selection ::= SEQUENCE
{
    cosem-attribute-descriptor         Cosem-Attribute-Descriptor,
    access-selection                   Selective-Access-Descriptor OPTIONAL
}

Get-Data-Result ::= CHOICE
{
   data                                [0] Data,
   data-access-result                  [1] IMPLICIT Data-Access-Result
}

Data-Block-Result ::= SEQUENCE  -- Used in ReadResponse with block transfer
{
    last-block                         BOOLEAN,
    block-number                       Unsigned16,
    raw-data                           OCTET STRING
}

DataBlock-G ::= SEQUENCE     -- G == DataBlock for the GET-response
{
    last-block                         BOOLEAN,
    block-number                       Unsigned32,
    result  CHOICE
    {
        raw-data                       [0] IMPLICIT OCTET STRING,
        data-access-result             [1] IMPLICIT Data-Access-Result
    }
}

DataBlock-SA ::= SEQUENCE    -- SA == DataBlock for the SET-request, ACTION-request and ACTION-response
{
    last-block                         BOOLEAN,
    block-number                       Unsigned32,
    raw-data                           OCTET STRING
}

Action-Response-With-Optional-Data ::= SEQUENCE
{
    result                             Action-Result,
    return-parameters                  Get-Data-Result  OPTIONAL
}

Notification-Body ::= SEQUENCE
{
   data-value                          Data
}

List-Of-Data ::= SEQUENCE OF Data
```

```
Access-Request-Get ::= SEQUENCE
{
   cosem-attribute-descriptor          Cosem-Attribute-Descriptor
}

Access-Request-Get-With-Selection ::= SEQUENCE
{
   cosem-attribute-descriptor          Cosem-Attribute-Descriptor,
   access-selection                    Selective-Access-Descriptor
}

Access-Request-Set ::= SEQUENCE
{
   cosem-attribute-descriptor          Cosem-Attribute-Descriptor
}

Access-Request-Set-With-Selection ::= SEQUENCE
{
   cosem-attribute-descriptor          Cosem-Attribute-Descriptor,
   access-selection                    Selective-Access-Descriptor
}

Access-Request-Action ::= SEQUENCE
{
   cosem-method-descriptor             Cosem-Method-Descriptor
}

Access-Request-Specification ::= CHOICE
{
   access-request-get                  [1] Access-Request-Get,
   access-request-set                  [2] Access-Request-Set,
   access-request-action               [3] Access-Request-Action,
   access-request-get-with-selection   [4] Access-Request-Get-With-Selection,
   access-request-set-with-selection   [5] Access-Request-Set-With-Selection
}

List-Of-Access-Request-Specification ::= SEQUENCE OF Access-Request-Specification

Access-Request-Body ::= SEQUENCE
{
   access-request-specification        List-Of-Access-Request-Specification,
   access-request-list-of-data         List-Of-Data
}

Access-Response-Get ::= SEQUENCE
{
   result                              Data-Access-Result
}

Access-Response-Set ::= SEQUENCE
{
   result                              Data-Access-Result
}

Access-Response-Action ::= SEQUENCE
{
   result                              Action-Result
}

Access-Response-Specification ::= CHOICE
{
   access-response-get                 [1] Access-Response-Get,
   access-response-set                 [2] Access-Response-Set,
   access-response-action              [3] Access-Response-Action
}

List-Of-Access-Response-Specification ::= SEQUENCE OF Access-Response-Specification

Access-Response-Body ::= SEQUENCE
{
   access-request-specification        [0] List-Of-Access-Request-Specification OPTIONAL,
   access-response-list-of-data        List-Of-Data,
   access-response-specification       List-Of-Access-Response-Specification
}
```

```
-- Key-info

Key-Id ::= ENUMERATED
{
        global-unicast-encryption-key      (0),
        global-broadcast-encryption-key    (1)
}

Kek-Id ::= ENUMERATED
{
        master-key                         (0)
}


Identified-Key ::= SEQUENCE
{
        key-id                             Key-Id
}

Wrapped-Key ::= SEQUENCE
{
    kek-id                             Kek-Id,
    key-ciphered-data                  OCTET STRING
}

Agreed-Key ::= SEQUENCE
{
    key-parameters                     OCTET STRING,
    key-ciphered-data                  OCTET STRING
}

Key-Info ::= CHOICE
{
    identified-key                     [0] Identified-Key,
    wrapped-key                        [1] Wrapped-Key,
    agreed-key                         [2] Agreed-Key
}

-- Use of Block-Control
--    window                  bits 0-5      window advertise
--    streaming               bit  6        0 = No Streaming active, 1 = Streaming active
--    last-block              bit  7        0 = Not Last Block, 1 = Last Block
Block-Control ::=                       Unsigned8


END
```