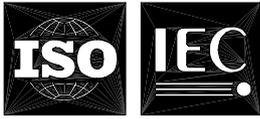ICS 35.100.70

ČSN
ISO/IEC 9596-1
OPRAVA 2
36 9646

Listopad 2004

ČESKÁ TECHNICKÁ NORMA

**Informační technologie – Propojení otevřených systémů –
Společný protokol informací managementu –
Část 1: Specifikace**

Information technology — Open Systems Interconnection — Common management information protocol —

Part 1:
Specification

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Protocole commun d'information de gestion —

Partie 1: Spécification

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to ISO/IEC 9596-1:1998 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

─────────────

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – COMMON MANAGEMENT INFORMATION PROTOCOL: SPECIFICATION

TECHNICAL CORRIGENDUM 2

Revision to include ASN.1: 1997

1) Subclause 2.1

Insert the following references into this subclause:

– ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

– ITU-T Recommendation X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.

– ITU-T Recommendation X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.

– ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

– ITU-T Recommendation X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

– ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.

– ITU-T Recommendation X.881 (1994) | ISO/IEC 13712-2:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) service definition*.

– ITU-T Recommendation X.882 (1994) | ISO/IEC 13712-3:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) protocol specification*.

2) Subclause 2.2

Replace subclause 2.2 with the following:

2.2 Paired Recommendations | International Standards equivalent in technical content

– CCITT Recommendation X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications*.

ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework*.

3) Subclause 3.3

Replace subclause 3.3 with the following:

3.3 Remote Operations definitions

This Recommendation | International Standard makes use of the following term defined in ITU-T Rec. X.880 | ISO/IEC 13172-1:

a)   linked-operation.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.881 | ISO/IEC 13172-2:

a)   association-initiator;

b)   association-responder;

c)   Remote Operation Service Element;

d)   invoker;

e)   performer.

4) Subclause 5.2

Replace the first paragraph with the following:

This Recommendation | International Standard uses the RO-INVOKE, RO-RESULT, RO-ERROR and RO-REJECT-U services of the Remote Operations Service Element (ROSE) defined in ITU-T Rec. X.880 | ISO/IEC 13712-1. ROSE assumes the use of the presentation service defined in ITU-T Rec. X.216 | ISO/IEC 8822. The confirmed operations of CMIP are asynchronous or synchronous as required by the application. The choice of asynchronous or synchronous is a local matter implemented by the application. The unconfirmed operations of CMIP are asynchronous, outcome not reported. Both the association-initiating and the association-responding application entities can invoke operations.

5) Subclause 5.2.2

Replace CCITT Rec. X.229 and ISO/IEC 9072-2 *with* ITU-T Rec. X.882 | ISO/IEC 13712-3.

6) Subclause 6.2.1

Replace CCITT Rec. X.229 and ISO/IEC 9072-2 *with* ITU-T Rec. X.882 | ISO/IEC 13712-3.

Replace InvokeID *with* invokeId *in Table 2*.

Replace Linked-ID *with* linkedId *in Table 2*.

7) Subclause 6.4.3

Replace linked-ID *with* linkedId *(occurs 4 times)*.

8) Subclause 6.4.4

Replace linked-ID *with* linkedId *(occurs twice)*.

9) Subclause 6.5.3

Replace linked-ID *with* linkedId *(occurs 4 times)*.

10) Subclause 6.5.4

Replace linked-ID *with* linkedId *(occurs twice)*.

11) Subclause 6.6.3

Replace linked-ID *with* linkedId *(occurs 4 times)*.

12)   Subclause 6.6.4

Replace linked-ID with linkedId *(occurs twice).*

13)   Subclause 6.8.3

Replace linked-ID *with* linkedId *(occurs 4 times).*

14)   Subclause 6.8.4

Replace linked-ID *with* linkedId *(occurs twice).*

15)   Subclause 7.1

Replace the first paragraph with the following:

The abstract syntax is defined using the notation specified in ITU-T Rec. X.680 | ISO/IEC 8824-1.

16)   Subclause 7.2

Replace Linked-ID *with* linkedId *in Table 4.*

Remove the last line of Table 4.

17)   Subclause 7.4

Replace with the following:

7.4   CMIP data units

The protocol is described in terms of Common Management Information Protocol Data Units exchanged between the peer CMISEs. The PDUs are specified using ASN.1 and the ROSE information objects defined in ITU-T Rec. X.880 | ISO/IEC 13712-1.

-- *Common Management Information Protocol (CMIP)*

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- *This ASN.1 specification has been checked for conformance with the ASN.1 standard by the OSS ASN.1 Tools*

IMPORTS

ERROR, OPERATION
    FROM Remote-Operations-Information-Objects
        {joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

ROS{}, InvokeId, noInvokeId
    FROM Remote-Operations-Generic-ROS-PDUs
        {joint-iso-itu-t remote-operations(4) generic-ROS-PDUs(6) version1(0)};

CMIP-Operations OPERATION ::= {
    m-Action |
    m-Action-Confirmed |
    m-CancelGet |
    m-Create |
    m-Delete |
    m-EventReport |
    m-EventReport-Confirmed |
    m-Get |
    m-Linked-Reply |
    m-Set |
    m-Set-Confirmed   }

```
CMIP-Confirmed-Operations OPERATION ::= {
     m-Action-Confirmed         |
     m-CancelGet                |
     m-Create                   |
     m-Delete                   |
     m-EventReport-Confirmed    |
     m-Get                      |
     m-Set-Confirmed            }
```

-- *CMISE error definitions*


```
accessDenied ERROR ::= {
     PRIORITY          {0}
     CODE              local : 2 }
```

```
classInstanceConflict ERROR ::= {
     PARAMETER    BaseManagedObjectId
     PRIORITY          {1}
     CODE              local : 19 }
```

```
complexityLimitation ERROR ::= {
     PARAMETER    ComplexityLimitation OPTIONAL TRUE
     PRIORITY          {1}
     CODE              local : 20 }
```

```
duplicateManagedObjectInstance ERROR ::= {
     PARAMETER    ObjectInstance
     PRIORITY          {1}
     CODE              local : 11 }
```

```
getListError ERROR ::= {
     PARAMETER    GetListError
     PRIORITY          {1}
     CODE              local : 7 }
```

```
invalidArgumentValue ERROR ::= {
     PARAMETER    InvalidArgumentValue
     PRIORITY          {1}
     CODE              local : 15 }
```

```
invalidAttributeValue ERROR ::= {
     PARAMETER    Attribute
     PRIORITY          {1}
     CODE              local : 6 }
```

```
invalidFilter ERROR ::= {
     PARAMETER    CMISFilter
     PRIORITY          {1}
     CODE              local : 4 }
```

```
invalidObjectInstance ERROR ::= {
     PARAMETER    ObjectInstance
     PRIORITY          {1}
     CODE              local : 17 }
```

```
invalidScope ERROR ::= {
     PARAMETER    Scope
     PRIORITY          {1}
     CODE              local : 16 }
```

```
missingAttributeValue ERROR ::= {
     PARAMETER    SET OF AttributeId
     PRIORITY          {1}
     CODE              local : 18 }
```

```
mistypedOperation ERROR ::= {
     PRIORITY          {1}
     CODE              local : 21 }
```

noSuchAction ERROR ::= {
    PARAMETER   **NoSuchAction**
    PRIORITY   **{1}**
    CODE   **local : 9 }**

noSuchArgument ERROR ::= {
    PARAMETER   **NoSuchArgument**
    PRIORITY   **{1}**
    CODE   **local : 14 }**

noSuchAttribute ERROR ::= {
    PARAMETER   **AttributeId**
    PRIORITY   **{1}**
    CODE   **local : 5 }**

noSuchEventType ERROR ::= {
    PARAMETER   **NoSuchEventType**
    PRIORITY   **{1}**
    CODE   **local : 13 }**

noSuchInvokeId ERROR ::= {
    PARAMETER   **InvokeIDType**
    PRIORITY   **{1}**
    CODE   **local : 22 }**

noSuchObjectClass ERROR ::= {
    PARAMETER   **ObjectClass**
    PRIORITY   **{1}**
    CODE   **local : 0 }**

noSuchObjectInstance ERROR ::= {
    PARAMETER   **ObjectInstance**
    PRIORITY   **{1}**
    CODE   **local : 1 }**

noSuchReferenceObject ERROR ::= {
    PARAMETER   **ObjectInstance**
    PRIORITY   **{1}**
    CODE   **local : 12 }**

operationCancelled ERROR ::= {
    PRIORITY   **{1}**
    CODE   **local : 23 }**

processingFailure ERROR ::= {
    PARAMETER   **ProcessingFailure OPTIONAL TRUE**
    PRIORITY   **{1}**
    CODE   **local : 10 }**

setListError ERROR ::= {
    PARAMETER   **SetListError**
    PRIORITY   **{1}**
    CODE   **local : 8 }**

syncNotSupported ERROR ::= {
    PARAMETER   **CMISSync**
    PRIORITY   **{1}**
    CODE   **local : 3 }**

-- *CMISE operations*

-- *Action operation (M-ACTION)*

m-Action OPERATION ::= {
    ARGUMENT   **ActionArgument**
    RETURN RESULT   **FALSE**
    ALWAYS RESPONDS   **FALSE**
    CODE   **local : 6 }**

m-Action-Confirmed OPERATION ::= {
    ARGUMENT   **ActionArgument**

RESULT	**ActionResult OPTIONAL TRUE**	*-- this result is conditional;*
		-- for conditions see 8.3.3.2.9 of ITU-T Rec. X.710
ERRORS	**{accessDenied | classInstanceConflict | complexityLimitation | invalidScope |**	
	invalidArgumentValue | invalidFilter | noSuchAction | noSuchArgument |	
	noSuchObjectClass | noSuchObjectInstance | processingFailure | syncNotSupported}	
LINKED	**{m-Linked-Reply}**	
CODE	**local : 7 }**	

-- Cancel get operation (M-CANCEL-GET)

m-CancelGet OPERATION ::= {

ARGUMENT	**InvokeIDType**	
RETURN RESULT	**TRUE**	
ERRORS	**{mistypedOperation | noSuchInvokeId | processingFailure}**	
CODE	**local : 10 }**	

-- Create operation (M-CREATE)

m-Create OPERATION ::= {

ARGUMENT	**CreateArgument**	
RESULT	**CreateResult OPTIONAL TRUE**	*-- this result is conditional;*
		-- for conditions see 8.3.4.1.3 of ITU-T Rec. X.710
ERRORS	**{accessDenied | classInstanceConflict | duplicateManagedObjectInstance |**	
	invalidAttributeValue | invalidObjectInstance | missingAttributeValue | noSuchAttribute |	
	noSuchObjectClass | noSuchObjectInstance | noSuchReferenceObject | processingFailure}	
CODE	**local : 8 }**	

-- Delete operation (M-DELETE)

m-Delete OPERATION ::= {

ARGUMENT	**DeleteArgument**	
RESULT	**DeleteResult OPTIONAL TRUE**	*-- this result is conditional;*
		-- for conditions see 8.3.5.2.8 of ITU-T Rec. X.710
ERRORS	**{accessDenied | classInstanceConflict | complexityLimitation | invalidFilter | invalidScope |**	
	noSuchObjectClass | noSuchObjectInstance | processingFailure | syncNotSupported}	
LINKED	**{m-Linked-Reply}**	
CODE	**local : 9 }**	

-- Event Reporting operations (M-EVENT-REPORT)

m-EventReport OPERATION ::= {

ARGUMENT	**EventReportArgument**	
RETURN RESULT	**FALSE**	
ALWAYS RESPONDS	**FALSE**	
CODE	**local : 0 }**	

m-EventReport-Confirmed OPERATION ::= {

ARGUMENT	**EventReportArgument**	
RESULT	**EventReportResult OPTIONAL TRUE**	
ERRORS	**{invalidArgumentValue | noSuchArgument | noSuchEventType |**	
	noSuchObjectClass | noSuchObjectInstance | processingFailure}	
CODE	**local : 1 }**	

-- Get operation (M-GET)

m-Get OPERATION ::= {

ARGUMENT	**GetArgument**	
RESULT	**GetResult OPTIONAL TRUE**	*-- this result is conditional;*
		-- for conditions see 8.3.1.2.8 of ITU-T Rec. X.710
ERRORS	**{accessDenied | classInstanceConflict | complexityLimitation | getListError |**	
	invalidFilter |	
	invalidScope | noSuchObjectClass | noSuchObjectInstance | operationCancelled|	
	processingFailure | syncNotSupported}	
LINKED	**{m-Linked-Reply}**	
CODE	**local : 3 }**	

-- Linked operation to M-GET, M-SET (Confirmed), M-ACTION (Confirmed), and M-DELETE

```
m-Linked-Reply OPERATION ::= {
    ARGUMENT          LinkedReplyArgument
    CODE                      local : 2 }
```

-- *Set operations (M-SET)*

```
m-Set OPERATION ::= {
    ARGUMENT          SetArgument
    RETURN RESULT     FALSE
    ALWAYS RESPONDS   FALSE
    CODE                      local : 4 }
```

```
m-Set-Confirmed OPERATION ::= {
    ARGUMENT          SetArgument
    RESULT                    SetResult OPTIONAL TRUE     -- *this result is conditional;*
                                                          -- *for conditions see 8.3.2.2.9 of ITU-T Rec. X.710*
    ERRORS                    {accessDenied | classInstanceConflict | complexityLimitation | invalidFilter |
                              invalidScope |
                              noSuchObjectClass | noSuchObjectInstance | processingFailure | setListError |
                              syncNotSupported}
    LINKED                    {m-Linked-Reply}
    CODE                      local : 5 }
```

-- *INFORMATION OBJECT definitions*
-- *While it is possible to use the Information object class definitions defined below to specify*
-- *Action types, Attribute types, Event Report types, and their associated ASN.1 type definitions,*
-- *the alternative approach using GDMO templates, as defined in ITU-T Rec. X.722 | ISO/IEC 10165-4,*
-- *continues to be available for use with this Recommendation | International Standard.*

```
CMIP-ACTION ::= CLASS {
    &id          ActionTypeId UNIQUE,
    &Value                           }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }
```

```
CMIP-ATTRIBUTE ::= CLASS {
    &id          AttributeId UNIQUE,
    &Value                           }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }
```

```
CMIP-AVA ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value                                 }
```

```
CMIP-EVENT ::= CLASS {
    &id          EventTypeId UNIQUE,
    &Value                           }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }
```

```
CMIP-SPECIFICERROR ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value                                 }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }
```

-- *Supporting type definitions*

```
AccessControl       ::= EXTERNAL
```

```
ActionArgument      ::= SEQUENCE {
    COMPONENTS OF     BaseManagedObjectId,
    accessControl             [5] AccessControl OPTIONAL,
    synchronization           [6] IMPLICIT CMISSync DEFAULT bestEffort,
    scope                     [7] Scope DEFAULT namedNumbers : baseObject,
```

```
    filter                      CMISFilter DEFAULT and : {},
    actionInfo                  [12] IMPLICIT ActionInfo,
    ...
    }

ActionError   ::= SEQUENCE {
    managedObjectClass          ObjectClass OPTIONAL,
    managedObjectInstance       ObjectInstance OPTIONAL,
    currentTime                 [5] IMPLICIT GeneralizedTime OPTIONAL,
    actionErrorInfo             [6] ActionErrorInfo,
    ...
    }

ActionErrorInfo     ::= SEQUENCE {
    errorStatus   ENUMERATED {accessDenied            (2),
                                noSuchAction            (9),
                                noSuchArgument          (14),
                                invalidArgumentValue    (15),
                                ... },
    errorInfo           CHOICE {
        actionType          CMIP-ACTION.&id ({ActionSet}),
        actionArgument   [0] NoSuchArgument,
        argumentValue    [1] InvalidArgumentValue
    },
    ... }

ActionInfo      ::= SEQUENCE {
    actionType          CMIP-ACTION.&id ({ActionSet}),
    actionInfoArg       [4] CMIP-ACTION.&Value ({ActionSet} {@.actionType}) OPTIONAL
    }

ActionReply    ::= SEQUENCE {
    actionType          CMIP-ACTION.&id ({ActionSet}),
    actionReplyInfo     [4] CMIP-ACTION.&Value ({ActionSet} {@.actionType})
    }

ActionResult   ::= SEQUENCE {
    managedObjectClass          ObjectClass OPTIONAL,
    managedObjectInstance       ObjectInstance OPTIONAL,
    currentTime                 [5] IMPLICIT GeneralizedTime OPTIONAL,
    actionReply                 [6] IMPLICIT ActionReply OPTIONAL,
    ...
    }

ActionSet CMIP-ACTION ::= {...}

ActionTypeId   ::= CHOICE {
    globalForm          [2] IMPLICIT OBJECT IDENTIFIER,
    localForm           [3] IMPLICIT INTEGER
    }
-- This Recommendation | International Standard does not allocate any values for localForm.
-- Where this alternative is used, the permissible values for the integers and their meanings shall be defined
-- as part of the application context in which they are used

Attribute  ::= SEQUENCE {
    id              CMIP-ATTRIBUTE.&id          ({AttributeSet}),
    value           CMIP-ATTRIBUTE.&Value       ({AttributeSet} {@.id})
    }

AttributeError ::= SEQUENCE {
    errorStatus                 ENUMERATED {    accessDenied                (2),
                                                noSuchAttribute             (5),
                                                invalidAttributeValue       (6),
                                                invalidOperation            (24),
                                                invalidOperator             (25),
                                                ... },
```

```
    modifyOperator    [2] IMPLICIT ModifyOperator OPTIONAL,        -- present for invalidOperator
                                                                   -- and invalidOperation
    attributeId       CMIP-ATTRIBUTE.&id ({AttributeSet}),
    attributeValue    CMIP-ATTRIBUTE.&Value ({AttributeSet} {@.attributeId}) OPTIONAL
                                                        -- value is absent for setToDefault
    }

AttributeId      ::= CHOICE {
    globalForm    [0] IMPLICIT OBJECT IDENTIFIER,
    localForm     [1] IMPLICIT INTEGER
    }
-- This Recommendation | International Standard does not allocate any values for localForm.
-- Where this alternative is used, the permissible values for the integers and their meanings shall be defined
-- as part of the application context in which they are used

AttributeIdError    ::= SEQUENCE {
    errorStatus   ENUMERATED {accessDenied       (2),
                              noSuchAttribute    (5),
                              ... },
    attributeId   AttributeId,
    ...
    }

AttributeSet CMIP-ATTRIBUTE ::= {...}

AttributeValueAssertion  ::= SEQUENCE {
    id        CMIP-AVA.&id        ({AvaSet}),
    value     CMIP-AVA.&Value     ({AvaSet} {@.id})
    }

AvaSet CMIP-AVA ::= {...}

BaseManagedObjectId    ::= SEQUENCE {
    baseManagedObjectClass        ObjectClass,
    baseManagedObjectInstance         ObjectInstance
    }

CMISFilter    ::= CHOICE {
    item    [8] FilterItem,
    and     [9] IMPLICIT SET OF CMISFilter,
    or      [10] IMPLICIT SET OF CMISFilter,
    not     [11] CMISFilter
    }

CMISSync      ::= ENUMERATED {        bestEffort    (0),
                                      atomic        (1) }

ComplexityLimitation     ::= SET {
    scope    [0] Scope OPTIONAL,
    filter[1] CMISFilter OPTIONAL,
    sync     [2] CMISSync OPTIONAL,
    ...
    }

CreateArgument     ::= SEQUENCE {
    managedObjectClass                       ObjectClass,
    managedOrSuperiorObjectInstance CHOICE {
        managedObjectInstance                ObjectInstance,
        superiorObjectInstance               [8] ObjectInstance } OPTIONAL,

accessControl                            [5] AccessControl OPTIONAL,
    referenceObjectInstance              [6] ObjectInstance OPTIONAL,
    attributeList                        [7] IMPLICIT SET OF Attribute OPTIONAL,
    ...
    }
```

```
CreateResult    ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,    -- shall be returned if omitted from CreateArgument
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList             [6] IMPLICIT SET OF Attribute OPTIONAL,
    ...
    }

DeleteArgument      ::= SEQUENCE {
    COMPONENTS OF    BaseManagedObjectId,
    accessControl           [5] AccessControl OPTIONAL,
    synchronization         [6] IMPLICIT CMISSync DEFAULT bestEffort,
    scope                   [7] Scope DEFAULT namedNumbers : baseObject,
    filter              CMISFilter DEFAULT and : {},
    ...
    }

DeleteError     ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    deleteErrorInfo           [6] ENUMERATED {  accessDenied       (2),
                                                  ... },
    ... }

DeleteResult    ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    ...
    }

DistinguishedName ::= RDNSequence

EventReply      ::= SEQUENCE {
    eventType          CMIP-EVENT.&id ({EventSet}),
    eventReplyInfo [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL
    }

EventReportArgument    ::= SEQUENCE {
    managedObjectClass        ObjectClass,
    managedObjectInstance     ObjectInstance,
    eventTime                 [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventType                 CMIP-EVENT.&id ({EventSet}),
    eventInfo                 [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL,
    ...
    }

EventReportResult  ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventReply                EventReply OPTIONAL,
    ...
    }

EventSet CMIP-EVENT ::= {...}

EventTypeId    ::= CHOICE {
    globalForm   [6] IMPLICIT OBJECT IDENTIFIER,
    localForm [7] IMPLICIT INTEGER
    }
```
-- *This Recommendation | International Standard does not allocate any values for localForm.*
-- *Where this alternative is used, the permissible values for the integers and their meanings shall be defined*
-- *as part of the application context in which they are used*

```
FilterItem        ::= CHOICE {
    equality              [0] IMPLICIT Attribute,
    substrings            [1] IMPLICIT SEQUENCE OF CHOICE {
        initialString         [0] IMPLICIT Attribute,
        anyString             [1] IMPLICIT Attribute,
        finalString           [2] IMPLICIT Attribute },
    greaterOrEqual        [2] IMPLICIT Attribute,    -- asserted value ≥ attribute value
    lessOrEqual           [3] IMPLICIT Attribute,    -- asserted value ≤ attribute value
    present               [4] AttributeId,
    subsetOf              [5] IMPLICIT Attribute,    -- asserted value is a subset of attribute value
    supersetOf            [6] IMPLICIT Attribute,    -- asserted value is a superset of attribute value
    nonNullSetIntersection[7] IMPLICIT Attribute
    }

GetArgument       ::= SEQUENCE {
    COMPONENTS OF    BaseManagedObjectId,
    accessControl         [5] AccessControl OPTIONAL,
    synchronization       [6] IMPLICIT CMISSync DEFAULT bestEffort,
    scope                 [7] Scope DEFAULT namedNumbers : baseObject,
    filter            CMISFilter DEFAULT and : {},
    attributeIdList       [12] IMPLICIT SET OF AttributeId OPTIONAL,
    ...
    }

GetInfoStatus     ::= CHOICE {
    attributeIdError   [0] IMPLICIT AttributeIdError,
    attribute          [1] IMPLICIT Attribute
    }

GetListError      ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    getInfoList               [6] IMPLICIT SET OF GetInfoStatus,
    ...
    }

GetResult         ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList             [6] IMPLICIT SET OF Attribute OPTIONAL,
    ...
    }

InvalidArgumentValue        ::= CHOICE {
    actionValue        [0] IMPLICIT ActionInfo,
    eventValue         [1] IMPLICIT SEQUENCE {
        eventType          CMIP-EVENT.&id ({EventSet}),
        eventInfo          [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL}
    }

InvokeIDType ::= InvokeId (ALL EXCEPT absent : NULL)

LinkedReplyArgument    ::= CHOICE {
    getResult             [0] IMPLICIT GetResult,
    getListError          [1] IMPLICIT GetListError,
    setResult             [2] IMPLICIT SetResult,
    setListError          [3] IMPLICIT SetListError,
    actionResult          [4] IMPLICIT ActionResult,
    processingFailure     [5] IMPLICIT ProcessingFailure,
    deleteResult          [6] IMPLICIT DeleteResult,
    actionError           [7] IMPLICIT ActionError,
    deleteError           [8] IMPLICIT DeleteError
    }
```

```
ModifyOperator    ::= INTEGER {    replace         (0),
                                   addValues       (1),
                                   removeValues    (2),
                                   setToDefault    (3) }

NoSuchAction      ::= SEQUENCE {
    managedObjectClass        ObjectClass,
    actionType                CMIP-ACTION.&id ({ActionSet}),
    ...
    }

NoSuchArgument    ::= CHOICE {
    actionId      [0] IMPLICIT SEQUENCE {
        managedObjectClass   ObjectClass OPTIONAL,
        actionType                CMIP-ACTION.&id ({ActionSet}) },
    eventId       [1] IMPLICIT SEQUENCE {
        managedObjectClass   ObjectClass OPTIONAL,
        eventType                 CMIP-EVENT.&id ({EventSet}) }
    }

NoSuchEventType   ::= SEQUENCE {
    managedObjectClass   ObjectClass,
    eventType                 CMIP-EVENT.&id ({EventSet}),
    ...
    }

ObjectClass    ::= CHOICE {
    globalForm   [0] IMPLICIT OBJECT IDENTIFIER,
    localForm    [1] IMPLICIT INTEGER
    }
```

-- *This Recommendation | International Standard does not allocate any values for localForm.*
-- *Where this alternative is used, the permissible values for the integers and their meanings shall be defined*
-- *as part of the application context in which they are used*

```
ObjectInstance    ::= CHOICE {
    distinguishedName           [2] IMPLICIT DistinguishedName,
    nonSpecificForm             [3] IMPLICIT OCTET STRING,
    localDistinguishedName      [4] IMPLICIT RDNSequence
    }
```

-- *localDistinguishedName is that portion of the distinguished name that is necessary to unambiguously identify the*
-- *managed object within the context of communication between the open systems*

```
ProcessingFailure    ::= SEQUENCE {
    managedObjectClass        ObjectClass,
    managedObjectInstance     ObjectInstance OPTIONAL,
    specificErrorInfo         [5] SpecificErrorInfo,
    ...
    }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeValueAssertion

Scope      ::= CHOICE { namedNumbers      INTEGER {    baseObject        (0),
                                                       firstLevelOnly    (1),
                                                       wholeSubtree      (2) },
    individualLevels        [1] IMPLICIT INTEGER,    -- POSITIVE integer indicates the level to be selected
    baseToNthLevel          [2] IMPLICIT INTEGER }   -- POSITIVE integer N indicates that the range of levels
                                                     -- (0-N) is to be selected
```

-- *with individualLevels and baseToNthLevel, a value of 0 has the same semantics as baseObject*
-- *with individualLevels, a value of 1 has the same semantics as firstLevelOnly*

```
SetArgument   ::= SEQUENCE {
    COMPONENTS OF        BaseManagedObjectId,
    accessControl        [5] AccessControl OPTIONAL,
    synchronization      [6] IMPLICIT CMISSync DEFAULT bestEffort,
    scope                [7] Scope DEFAULT namedNumbers : baseObject,
    filter               CMISFilter DEFAULT and : {},
    modificationList     [12] IMPLICIT SET OF SEQUENCE {
        modifyOperator      [2] IMPLICIT ModifyOperator DEFAULT replace,
        attributeId         CMIP-ATTRIBUTE.&id ({AttributeSet}),
        attributeValue      CMIP-ATTRIBUTE.&Value ({AttributeSet} {@.attributeId}) OPTIONAL },
                                            -- value is absent for setToDefault
    ... }

SetInfoStatus         ::= CHOICE {
    attributeError    [0] IMPLICIT AttributeError,
    attribute         [1] IMPLICIT Attribute
    }

SetListError   ::= SEQUENCE {
    managedObjectClass        ObjectClass OPTIONAL,
    managedObjectInstance     ObjectInstance OPTIONAL,
    currentTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    setInfoList               [6] IMPLICIT SET OF SetInfoStatus,
    ...
    }

SetResult      ::= SEQUENCE {
    managedObjectClass         ObjectClass OPTIONAL,
    managedObjectInstance      ObjectInstance OPTIONAL,
    currentTime                [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList              [6] IMPLICIT SET OF Attribute OPTIONAL,
    ...
    }

SpecificErrorInfo ::= SEQUENCE {
    errorId      CMIP-SPECIFICERROR.&id ({SpecificErrorSet}),
    errorInfo    CMIP-SPECIFICERROR.&Value ({SpecificErrorSet} {@.errorId})
    }

SpecificErrorSet CMIP-SPECIFICERROR ::= {...}
```

-- *the following type specifies the constraints to be applied when using ROSE to support CMIP*

ROSEapdus ::= ROS{{InvokeIDType}, {CMIP-Operations}, {CMIP-Confirmed-Operations}}

END   -- *End of CMIP syntax definitions*

18)   Subclause 7.5

Replace Remote-Operations-APDUs.ROSEapdus *with* Remote-Operations-Generic-ROS-PDUs.ROS.

Replace CCITT Rec. X.229 and ISO/IEC 9072-2 *with* ITU-T Rec. X.880 | ISO/IEC 13712-1.

Replace CCITT Rec. X.209 and ISO/IEC 8825 *with* ITU-T Rec. X.690 | ISO/IEC 8825-1.

19)   Subclause 8.1

Replace CCITT Rec. X.209 and ISO/IEC 8825 *with* ITU-T Rec. X.690 | ISO/IEC 8825-1.

Replace CCITT Rec. X.229 and ISO/IEC 9072-2 *with* ITU-T Rec. X.880 | ISO/IEC 13712-1.

Replace item f) with the following:

> f)   support the ability of both the association-initiating and the association-responding application entities to invoke operations;

20) Annex B

Replace with the following:

Annex B

Expanded ASN.1 syntax

(This annex does not form an integral part of this Recommendation | International Standard)

This annex describes how the OPERATION and ERROR information objects of ITU-T Rec. X.880 | ISO/IEC 13712-1 are expanded into ASN.1 data types and subtypes.

If any inconsistencies exist between these definitions and the definitions in clause 7, then the definitions in clause 7 take precedence.

-- *Common Management Information Protocol (CMIP)*

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- *This ASN.1 specification has been checked for conformance with the ASN.1 standard by the OSS ASN.1 Tools*

IMPORTS

ERROR, OPERATION
        FROM Remote-Operations-Information-Objects
            {joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

ROS{}, InvokeId, noInvokeId
        FROM Remote-Operations-Generic-ROS-PDUs
            {joint-iso-itu-t remote-operations(4) generic-ROS-PDUs(6) version1(0)};

```
CMIP-Operations OPERATION ::=  {
    m-Action                |
    m-Action-Confirmed      |
    m-CancelGet             |
    m-Create                |
    m-Delete                |
    m-EventReport           |
    m-EventReport-Confirmed |
    m-Get                   |
    m-Linked-Reply          |
    m-Set                   |
    m-Set-Confirmed         }
CMIP-Confirmed-Operations OPERATION ::= {
    m-Action-Confirmed      |
    m-CancelGet             |
    m-Create                |
    m-Delete                |
    m-EventReport-Confirmed |
    m-Get                   |
    m-Set-Confirmed         }
```

-- *INFORMATION OBJECT definitions*
-- *While it is possible to use the Information object class definitions defined below to specify*
-- *Action types, Attribute types, Event Report types, and their associated ASN.1 type definitions,*
-- *the alternative approach using GDMO templates, as defined in ITU-T Rec. X.722 | ISO/IEC 10165-4,*
-- *continues to be available for use with this Recommendation | International Standard.*

```
CMIP-ACTION ::= CLASS {
    &id         ActionTypeId UNIQUE,
    &Value                      }
    WITH SYNTAX  {TYPE   &Value
                    ID      &id }
```

```
CMIP-ATTRIBUTE ::= CLASS {
    &id          AttributeId UNIQUE,
    &Value                          }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }

CMIP-AVA ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value                               }

CMIP-EVENT ::= CLASS {
    &id          EventTypeId UNIQUE,
    &Value                          }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }

CMIP-SPECIFICERROR ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value                               }
    WITH SYNTAX  {TYPE   &Value
                  ID     &id }
```
-- *the following type specifies the constraints to be applied when using ROSE to support CMIP*

ROSEapdus ::= ROS{{InvokeIDType}, {CMIP-Operations}, {CMIP-Confirmed-Operations}}
-- *CMISE operations*
-- *The following part of the ASN.1 specification provides a definition of ROIVapdu and RORSapdu subtypes used by CMIP.*
-- *The subtypes of the ROIVapdu define the allowed values of the operation-value and argument defined by that*
-- *operation-value for all CMIP notifications and operations. The subtypes of the RORSapdu define the allowed*
-- *values of the operation-value and result defined by that operation-value for all CMIP notifications and operations.*

m-Action OPERATION.&operationCode ::= local : 6

```
ROIV-m-Action ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Action),
        argument      (ActionArgument) } )
    })
```

m-Action-Confirmed OPERATION.&operationCode ::= local : 7

```
ROIV-m-Action-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Action-Confirmed),
        argument      (ActionArgument) } )
    })
```

```
RORS-m-Action-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode        (m-Action-Confirmed),
            result        (ActionResult) ) ) OPTIONAL } )
            -- required only if there is a single reply to the ROIV-m-Action-Confirmed ROSEapdu
            -- and data is to be returned in the ROSEapdu
    } )
```

m-Cancel-Get OPERATION.&operationCode ::= local : 10

```
ROIV-m-Cancel-Get ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Cancel-Get),
        argument      (InvokeIDType) } )
    } )
```

RORS-m-Cancel-Get ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId **(InvokeIDtype) })**
    -- There is no result sequence for RORS-m-Cancel-Get
  })

m-Create OPERATION.&operationCode ::= local : 8

ROIV-m-Create ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    linkedId **ABSENT,**
    opcode **(m-Create),**
    argument **(CreateArgument) })**
  })

RORS-m-Create ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    result **(WITH COMPONENTS {**
      opcode **(m-Create),**
      result **(CreateResult) }) })**
  })

m-Delete OPERATION.&operationCode ::= local : 9

ROIV-m-Delete ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    linkedId **ABSENT,**
    opcode **(m-Delete),**
    argument **(DeleteArgument) })**
  })

RORS-m-Delete ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    result **(WITH COMPONENTS {**
      opcode **(m-Delete),**
      result **(DeleteResult) }) OPTIONAL })**
    -- required only if there is a single reply to the ROIV-m-DeleteROSEapdu
    -- and data is to be returned in the ROSEapdu
  })

m-EventReport OPERATION.&operationCode ::= local : 0

ROIV-m-EventReport ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    linkedId **ABSENT,**
    opcode **(m-EventReport),**
    argument **(EventReportArgument) })**
  })

m-EventReport-Confirmed OPERATION.&operationCode ::= local : 1

ROIV-m-EventReport-Confirmed ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    linkedId **ABSENT,**
    opcode **(m-EventReport-Confirmed),**
    argument **(EventReportArgument) })**
  })

RORS-m-EventReport-Confirmed ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId **(InvokeIDtype),**
    result   **(WITH COMPONENTS {**

```
        opcode      (m-EventReport-Confirmed),
        result      (EventReportResult) } ) OPTIONAL } )
            -- required only if data is to be returned in the ROSEapdu
    } )

m-Get OPERATION.&operationCode ::= local : 3

ROIV-m-Get ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Get),
        argument      (GetArgument) } )
    } )

RORS-m-Get ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode        (m-Get),
            result        (GetResult) } ) OPTIONAL } )
            -- required only if there is a single reply to the ROIV-m-Get ROSEapdus
    } )

m-Linked-Reply OPERATION.&operationCode ::= local : 2

ROIV-m-Linked-Reply ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      PRESENT,
        opcode        (m-Linked-Reply),
        argument      (LinkedReplyArgument) } )
    } )
-- This part of the ASN.1 specification provides a definition of ROIV-m-Linked-Reply subtypes used by CMIP.
-- The subtypes of the ROIV-m-Linked-Reply ROSEapdus define the allowed values of the argument defined by
-- the opcode for the specific CMIP linked reply operations.

ROIV-m-Linked-Reply-Action ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      PRESENT,
        opcode        (m-Linked-Reply),
        argument      (LinkedReplyArgument (WITH COMPONENTS {
            invoke (WITH COMPONENTS {
                getResult        ABSENT,
                getListError     ABSENT,
                setResult        ABSENT,
                setListError     ABSENT,
                actionResult     PRESENT,
                processingFailure PRESENT,
                deleteResult     ABSENT,
                actionError      PRESENT,
                deleteError      ABSENT } )
        } ) )
    } ) } )

ROIV-m-Linked-Reply-Delete ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      PRESENT,
        opcode        (m-Linked-Reply),
        argument      (LinkedReplyArgument (WITH COMPONENTS {
            invoke (WITH COMPONENTS {
                getResult        ABSENT,
                getListError     ABSENT,
                setResult        ABSENT,
                setListError     ABSENT,
```

```
                    actionResult       ABSENT,
                    processingFailure  PRESENT,
                    deleteResult       PRESENT,
                    actionError        ABSENT,
                    deleteError        PRESENT } )
             } ) )
      } ) } )

ROIV-m-Linked-Reply-Get ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
      invoke (WITH COMPONENTS {
             invokeId        (InvokeIDtype),
             linkedId        PRESENT,
             opcode          (m-Linked-Reply),
             argument        (LinkedReplyArgument (WITH COMPONENTS {
                    invoke (WITH COMPONENTS {
                           getResult          PRESENT,
                           getListError       PRESENT,
                           setResult          ABSENT,
                           setListError       ABSENT,
                           actionResult       ABSENT,
                           processingFailure  PRESENT,
                           deleteResult       ABSENT,
                           actionError        ABSENT,
                           deleteError        ABSENT } )
             } ) )
      } ) } )

ROIV-m-Linked-Reply-Set ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
      invoke (WITH COMPONENTS {
             invokeId        (InvokeIDtype),
             linkedId        PRESENT,
             opcode          (m-Linked-Reply),
             argument        (LinkedReplyArgument (WITH COMPONENTS {
                    invoke (WITH COMPONENTS {
                           getResult          ABSENT,
                           getListError       ABSENT,
                           setResult          PRESENT,
                           setListError       PRESENT,
                           actionResult       ABSENT,
                           processingFailure  PRESENT,
                           deleteResult       ABSENT,
                           actionError        ABSENT,
                           deleteError        ABSENT } )
             } ) )
      } ) } )

m-Set OPERATION.&operationcode ::= local : 4

ROIV-m-Set ::= ROSEapdus (WITH COMPONENTS {
      invoke (WITH COMPONENTS {
             invokeId        (InvokeIDtype),
             linkedId        ABSENT,
             opcode          (m-Set),
             argument        (SetArgument) } )
      } )

m-Set-Confirmed OPERATION.&operationCode ::= local : 5

ROIV-m-Set-Confirmed ::= ROSEapdus (WITH COMPONENTS {
      invoke (WITH COMPONENTS {
             invokeId        (InvokeIDtype),
             linkedId        ABSENT,
             opcode          (m-Set-Confirmed),
             argument        (SetArgument) } )
      } )
```

```
RORS-m-Set-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode      (m-Set-Confirmed),
            result       (SetResult) } ) OPTIONAL } )
            -- required only if there is a single reply to the ROIV-m-Set-Confirmed ROSEapdu
            -- and data is to be returned in the ROSEapdu
    } )
```

-- The following part of the ASN.1 specification provides a definition of ROERapdu subtypes used by CMIP.
-- The subtypes of the ROERapdu define the allowed values of the error value and parameter defined by that
-- error-value for all CMIP notifications and operations.

accessDenied ERROR.&errorCode ::= local : 2

```
ROER-accessDenied ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (accessDenied) } )
```
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create and ROIV-m-Delete ROIVapdus
```
    } )
```

classInstanceConflict ERROR.&errorCode ::= local : 19

```
ROER-classInstanceConflict ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (classInstanceConflict),
        parameter     (INCLUDES BaseManagedObjectId) } )
```
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create and ROIV-m-Delete ROIVapdus
```
    } )
```

complexityLimitation ERROR.&errorCode ::= local : 20

```
ROER-complexityLimitation ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (complexityLimitation),
        parameter     (INCLUDES ComplexityLimitation) OPTIONAL } )
```
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus
```
    } )
```

duplicateManagedObjectInstance ERROR.&errorCode ::= local : 11

```
ROER-duplicateManagedObjectInstance ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (duplicateManagedObjectInstance),
        parameter     (INCLUDES ObjectInstance) } )
```
-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu
```
    } )
```

getListError ERROR.&errorCode ::= local : 7

```
ROER-getListError ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (getListError),
        parameter     (INCLUDES GetListError) } )
```
-- This ROERapdu may only be returned in response to the ROIV-m-Get ROIVapdu
```
    } )
```

invalidArgumentValue ERROR.&errorCode ::= local : 15

ROER-invalidArgumentValue ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (invalidArgumentValue),
        parameter   (INCLUDES InvalidArgumentValue) })
-- This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed
-- and ROIV-m-Action-Confirmed ROIVapdus
    })

invalidAttributeValue ERROR.&errorCode ::= local : 6

ROER-invalidAttributeValue ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (invalidAttributeValue),
        parameter   (INCLUDES Attribute) })
-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu
    })

invalidFilter ERROR.&errorCode ::= local : 4

ROER-invalidFilter ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (invalidFilter),
        parameter   (INCLUDES CMISFilter) })
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus
    })

invalidObjectInstance ERROR.&errorCode ::= local : 17

ROER-invalidObjectInstance ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (invalidObjectInstance),
        parameter   (INCLUDES ObjectInstance) })
-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu
    })

invalidScope ERROR.&errorCode ::= local : 16

ROER-invalidScope ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (invalidScope),
        parameter   (INCLUDES Scope) })
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus
    })

missingAttributeValue ERROR.&errorCode ::= local : 18

ROER-missingAttributeValue ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (missingAttributeValue),
        parameter   (INCLUDES SET OF AttributeId) })
-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu
    })

mistypedOperation ERROR.&errorCode ::= local : 21

ROER-mistypedOperation ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (mistypedOperation) })
-- This ROERapdu may only be returned in response to the ROIV-m-Cancel-Get ROIVapdu
    })

noSuchAction ERROR.&errorCode ::= local : 9

ROER-noSuchAction ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchAction),
        parameter   (INCLUDES NoSuchAction) })
-- This ROERapdu may only be returned in response to the ROIV-m-Action-Confirmed ROIVapdu
    })

noSuchArgument ERROR.&errorCode ::= local : 14

ROER-noSuchArgument ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchArgument),
        parameter   (INCLUDES NoSuchArgument) })
-- This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed and
-- ROIV-m-Action-Confirmed ROIVapdus
    })

noSuchAttribute ERROR.&errorCode ::= local : 5

ROER-noSuchAttribute ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchAttribute),
        parameter   (INCLUDES AttributeId) })
-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu
    })

noSuchEventType ERROR.&errorCode ::= local : 13

ROER-noSuchEventType ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchEventType),
        parameter   (INCLUDES NoSuchEventType) })
-- This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed ROIVapdu
    })

noSuchInvokeId ERROR.&errorCode ::= local : 22

ROER-noSuchInvokeId ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchInvokeId),
        parameter   (INCLUDES InvokeIDType) })
-- This ROERapdu may only be returned in response to the ROIV-m-Cancel-Get ROIVapdu
    })

noSuchObjectClass ERROR.&errorCode ::= local : 0

ROER-noSuchObjectClass ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchObjectClass),
        parameter   (INCLUDES ObjectClass) })
-- This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get,
-- ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus
    })

noSuchObjectInstance ERROR.&errorCode ::= local : 1

ROER-noSuchObjectInstance ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId   PRESENT,
        errcode   (noSuchObjectInstance),
        parameter   (INCLUDES ObjectInstance) })

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get,*
-- *ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus*
  })

noSuchReferenceObject ERROR.&errorCode ::= local : 12

ROER-noSuchReferenceObject ::= ROSEapdus (WITH COMPONENTS {
  returnError (WITH COMPONENTS {
    invokeId PRESENT,
    errcode (noSuchReferenceObject),
    parameter (INCLUDES ObjectInstance) })
-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*
  })

operationCancelled ERROR.&errorCode ::= local : 23

ROER-operationCancelled ::= ROSEapdus (WITH COMPONENTS {
  returnError (WITH COMPONENTS {
    invokeId PRESENT,
    errcode (operationCancelled) })
-- *This ROERapdu may only be returned in response to the ROIV-m-Get ROIVapdu*
  })

processingFailure ERROR.&errorCode ::= local : 10

ROER-processingFailure ::= ROSEapdus (WITH COMPONENTS {
  returnError (WITH COMPONENTS {
    invokeId PRESENT,
    errcode (processingFailure),
    parameter (INCLUDES ProcessingFailure) OPTIONAL })
-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get,*
-- *ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus*
  })

setListError ERROR.&errorCode ::= local : 8

ROER-setListError ::= ROSEapdus (WITH COMPONENTS {
  returnError (WITH COMPONENTS {
    invokeId PRESENT,
    errcode (setListError),
    parameter (INCLUDES SetListError) })
-- *This ROERapdu may only be returned in response to the ROIV-m-Set-Confirmed ROIVapdu*
  })

syncNotSupported ERROR.&errorCode ::= local : 3

ROER-syncNotSupported ::= ROSEapdus (WITH COMPONENTS {
  returnError (WITH COMPONENTS {
    invokeId PRESENT,
    errcode (syncNotSupported),
    parameter (INCLUDES CMISSync) })
-- *This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,*
-- *ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus*
  })

-- *To complete the abstract syntax specification provided in this annex, the definitions of the supporting types in 7.4*
-- *are incorporated by reference*

END -- *of CMIP syntax definitions*

21) Annexes C and D

Remove both annexes.