INTERNATIONAL STANDARD

ISO/IEC 9506-2

First edition
1990-10-15

AMENDMENT 2
1995-12-15

Industrial automation systems — Manufacturing Message Specification —

Part 2:
Protocol specification

AMENDMENT 2: Conditioned service response

Systèmes d'automatisation industrielle — Spécification de messagerie industrielle —

Partie 2: Spécification de protocole

AMENDEMENT 2: Réponse conditionnelle de service

Reference number
ISO/IEC 9506-2:1990/Amd.2:1995(E)

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Amendment 2 to International Standard ISO/IEC 9506-2:1990 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee 5, *Architecture and communications*.

Introduction

This amendment details the changes to ISO/IEC 9506-2 to support conditioned service response. In developing these changes, it is assumed that the changes from the inclusion of the Data Exchange Service, ISO/IEC 9506-2/Amd.1, and the changes from technical corrigendum ISO/IEC 9506-2/Cor.1, have already been applied to the base document. All clause number references refer to the document as amended and corrected; page number references refer to the base document.

This amendment adds a new object, an Access Control List, to the structure of the MMS VMD. The VMD references one such object that provides conditions that constrain the successful access of any object within the VMD by an MMS Client. In addition, each named object within a VMD references some Access Control List object, and the conditions expressed in that Access Control List object constrain the use of the parent object by an MMS Client. The present MMS system allows an MMS Server to support or deny support for any MMS service to an MMS Client for all object instances within its implementation; this amendment allows an MMS Server to offer support for a MMS service to an MMS Client for some object instances but not for others. If support of object specific access control is negotiated in the Initiate dialogue, the MMS client may examine and manipulate the Access Control List object of individual object instances.

The attribute MMS Deletable is removed from the object description of all MMS objects. In its place, a derivation rule is provided such that services that report MMS Deletable can do so in a manner consistent with implementations not employing this amendment.

There are seven classes of constraint, called Service Classes, that are covered by this amendment. These classes are Read, Write, Load, Store, Execute, Delete, and Edit. Not all classes are applicable to all objects. The Edit class describes the ability to change the Access Control List characteristics of any object.

This amendment makes use of the Authentication Unit of the Association Control Service Element (ACSE) now available as an implementation option. It does so by allowing the conditions expressed in the Access Control List to depend on the Authentication Value present in the A-ASSOCIATE service primitives. Such use of the Authentication Unit is not required, however, to make use of the Access Control List mechanism.

By using the mechanisms present in this amendment, an implementation can restrict access to an object (for reading, writing, loading, storing, execution, deletion, or other modification) to MMS Clients that either (1) attempt access from known network nodes, (2) provide properauthentication (passwords), (3) have synchronized their use with other MMS Clients through use of the semaphores, or (4) an arbitrary combination of these methods. The specification of passwords requires the use of the Authentication Unit of ACSE.

This amendment also modifies the MMS Service model by adding an explicit Object Model for an Application Association. This model should be present in the basic MMS Object Model, independent of the use of Access Control Lists. Its omission in the base document should be considered an oversight, corrected by this amendment.

The introduction of an object model for the application association allows one to move the list of transactions objects from the VMD to the application association, thereby allowing the invoke ID to be the sole key attribute of the transaction. The case of processing of Event Actions, however, requires us to introduce a new attribute to the VMD, namely a list of transactions associated with Event Action processing that are not bound (necessarily) to an association.

Industrial automation systems — Manufacturing Message Specification —

Part 2:
Protocol specification

AMENDMENT 2: Conditioned service response

1. Scope

(This amendment makes no changes to clause 1 of ISO/IEC 9506-2.)

2. Normative References

Immediately following the reference to ISO 8650, page 2, add the following:

ISO 8649:1988/Amd. 1: 1990, *Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element Amendment 1: Authentication during association establishment.*

ISO 8650:1988/Amd 1:1990, *Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element Amendment 1: Authentication during association establishment.*

Immediately following the reference to ISO/IEC 9506-1, add the following:

ISO/IEC 9506-2:1990/Amd. 1:1993, *Industrial automation systems - Manufacturing Message Specification - Part 2: Protocol Specification Amendment 1: Data Exchange.*

ISO/IEC 9506-2:1990/Cor. 1:1995, *Industrial Automation Systems - Manufacturing Message Specification - Part 2: Protocol Specification Technical corrigendum 1.*

3. Definitions

(This amendment makes no changes to clause 3 of ISO/IEC 9506-2.)

4. Abbreviations

(This amendment makes no changes to clause 4 of ISO/IEC 9506-2.)

5. Conventions

(This amendment makes no changes to clause 5 of ISO/IEC 9506-2.)

6. Elements of Protocol Procedure

(This amendment makes no changes to clause 6 of ISO/IEC 9506-2.)

7. MMS PDU

In clause 7, near the top of page 19, replace the lines

```
IMPORTS AP-title, AP-invocation-id, AE-qualifier, AE-invocation-id
  FROM ISO-8650-ACSE-1
{iso standard 8650 abstract-syntax(2) acse-pdi(1)};
```

with:

```
IMPORTS AP-title, AP-invocation-id, AE-qualifier,
   AE-invocation-id, Authentication-value
   FROM ISO-8650-ACSE-1
{iso standard 8650 abstract-syntax(2) acse-pdi(1)};
```

In 7.5.2, in the middle of page 23, change the last line of the production from:

```
  exchangeData                [81] IMPLICIT ExchangeData-Request
                        -- Shall not appear in minor version 1
```

to:

```
 exchangeData          [81] IMPLICIT ExchangeData-Request,
                -- Shall not appear in minor version one
 defineAccessControlList  [82] IMPLICIT
                   DefineAccessControlList-Request,
               -- Shall not appear in minor version one or two
 getAccessControlListAttributes   [83] IMPLICIT
                   GetAccessControlListAttributes-Request,
                -- Shall not appear in minor version one or two
 reportAccessControlledObjects    [84] IMPLICIT
                   ReportAccessControlledObjects-Request,
                -- Shall not appear in minor version one or two
 deleteAccessControlList          [85] IMPLICIT
                   DeleteAccessControlList-Request,
                -- Shall not appear in minor version one or two
 changeAccessControl              [86] IMPLICIT
                   ChangeAccessControl-Request
                -- Shall not appear in minor version one or two
```

In 7.5.4 near the top of page 27, change the last line of the production from:

```
exchangeData                        [81] IMPLICIT
                        ExchangeData-Response
                    -- Shall not appear in minor version one
```

to

```
exchangeData                        [81] IMPLICIT
                        ExchangeData-Response,
                    -- Shall not appear in minor version one
defineAccessControlList             [82] IMPLICIT
                        DefineAccessControlList-Response,
                    -- Shall not appear in minor version one or two
getAccessControlListAttributes      [83] IMPLICIT
                        GetAccessControlListAttributes-Response,
                    -- Shall not appear in minor version one or two
reportAccessControlledObjects       [84] IMPLICIT
                        ReportAccessControlledObjects-Response,
                    -- Shall not appear in minor version one or two
deleteAccessControlList             [85] IMPLICIT
                        DeleteAccessControlList-Response,
                    -- Shall not appear in minor version one or two
changeAccessControl                 [86] IMPLICIT
                        ChangeAccessControl-Response
                    -- Shall not appear in minor version one or two
```

In 7.5.5, page 30, at the very end of the ServiceError production, change:

```
    additionalService     [10] AdditionalService-Error
    } OPTIONAL
```

to:

```
    additionalService     [10] AdditionalService-Error,
    changeAccessControl   [11] IMPLICIT ChangeAccessControlError
                        -- Shall not appear in minor version one or two
    } OPTIONAL
```

8. Environment and General Management Protocol

In 8.2 on page 36, change:

```
    cancel                                          (84),
    getDataExchangeAttributes                       (85),
       -- Shall not appear in minor version one
    exchangeData                                    (86)
       -- Shall not appear in minor version one
```

to:

```
    cancel                                          (84),
    getDataExchangeAttributes                       (85),
       -- Shall not appear in minor version one
    exchangeData                                    (86),
       -- Shall not appear in minor version one
    defineAccessControlList                         (87),
       -- Shall not appear in minor version one or two
    getAccessControlListAttributes                  (88),
       -- Shall not appear in minor version one or two
    reportAccessControlledObjects                   (89),
       -- Shall not appear in minor version one or two
    deleteAccessControlList                         (90),
       -- Shall not appear in minor version one or two
    alterAccessControl                              (91)
       -- Shall not appear in minor version one or two
```

In 8.2 on page 36, in ParameterSupportOptions, change:

```
   cei    (10)
```

to:

```
   cei    (10),
   aco    (11)       -- Shall not appear in minor version one or two
```

9. VMD Support Protocol

In 9.4 on page 41, change:

```
    operatorStation                                 (11),
    dataExchange                                    (12)
       -- Shall not appear in minor version one
```

to:

```
    operatorStation                              (11),
    dataExchange                                 (12),
       -- Shall not appear in minor version one
    accessControlList                            (13)
       -- Shall not appear in minor version one or two
```

In 9.6 on page 43, change:

```
    operatorStation                              (11),
    dataExchange                                 (12)
       -- Shall not appear in minor version one
```

to:

```
    operatorStation                              (11),
    dataExchange                                 (12),
       -- Shall not appear in minor version one
    accessControlList                            (13)
       -- Shall not appear in minor version one or two
```

10. Domain Management Protocol

In 10.13, page 50, replace the line:

```
      uploadInProgress       [5] IMPLICIT Integer8
```

with:

```
      uploadInProgress  [5] IMPLICIT Integer8,
      accessControlList [6] IMPLICIT Identifier OPTIONAL
                  -- Shall not appear in minor version one or two
```

Add a new subclause 10.13.2.1:

10.13.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

11. Program Invocation Management Protocol

In 11.9, bottom of page 55, replace the last five lines with:

```
executionArgument       CHOICE {
    simpleString            [5] IMPLICIT VisibleString,
    encodedString           EXTERNAL
    },
accessControlList       [6] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two
}
```

Add a new subclause 11.9.2.2:

11.9.2.2 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

12. Variable Access Protocol

In 12.9, bottom of page 64, replace the last two lines with:

```
typeSpecification    [2] TypeSpecification,
accessControlList    [3] IMPLICIT Identifier OPTIONAL
             -- Shall not appear in minor version one or two
}
```

Add a new subclause 12.9.2.1:

12.9.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

In 12.12, page 66, replace the last two lines with:

```
scatteredAccessDescription   [1] IMPLICIT
                                 ScatteredAccessDescription,
accessControlList        [2] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two
}
```

Add a new subclause 12.12.2.1:

12.12.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

In 12.15, page 67, replace the last five lines with:

```
listOfVariable      [1] IMPLICIT SEQUENCE OF SEQUENCE {
  variableSpecification   VariableSpecification,
  alternateAccess      [5] IMPLICIT AlternateAccess OPTIONAL
  },
accessControlList    [2] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two      }
```

Add a new subclause 12.15.2.1:

12.15.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

In 12.18, page 69, replace the last two lines with:

```
typeSpecification    TypeSpecification,
accessControlList    [1] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two        }
```

Add a new subclause 12.18.2.1:

12.18.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

13. Semaphore Management Protocol

In 13.6, page 72, replace the last two lines with:

```
numberOfHungTokens      [4] IMPLICIT Unsigned16,
accessControlList       [5] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two
}
```

Add a new subclause 13.6.2.1:

13.6.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

14. Operator Communication Protocol

(This amendment makes no changes to clause 14 of ISO/IEC 9506-2.)

15. Event Management Protocol

In 15.4, page 77, replace the last two lines with:

```
evaluationInterval      [7] IMPLICIT Unsigned32 OPTIONAL,
accessControlList       [8] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two
}
```

Add a new subclause 15.4.2.3:

15.4.2.3 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

In 15.10, page 81, replace the last four lines with:

```
cs-extension          [79] CS-Request-Detail OPTIONAL,
       -- shall not be transmitted if value is the value
       -- of a tagged type derived from NULL
accessControlList     [3] IMPLICIT Identifier OPTIONAL
           -- Shall not appear in minor version one or two
}
```

Add a new section 15.10.2.2:

15.10.2.2 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

In 15.14, page 84, replace the last three lines with:

```
additionalDetail  [9] IMPLICIT EE-Additional-Detail OPTIONAL,
        -- shall not be transmitted if the value is NULL
accessControlList       [11] IMPLICIT Identifier OPTIONAL
       -- Shall not appear in minor version one or two
 }
```

Add a new subclause 15.14.2.1.5:

15.14.2.1.5 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

16. Journal Management Protocol

In 16.5, page 95, replace the last three lines with:

```
currentEntries          [0] IMPLICIT Unsigned32,
mmsDeletable            [1] IMPLICIT BOOLEAN,
accessControlList       [2] IMPLICIT Identifier OPTIONAL
            -- Shall not appear in minor version one or two
}
```

Add a new subclause 16.5.2.1:

16.5.2.1 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

17. Mapping to ACSE and Presentation Services

Insert a new subclause 17.2:

17.2 A-ASSOCIATE Data

Data from the A-ASSOCIATE Request or Response services shall be passed for the initialization of attributes of the Application Association Object that is created for the association. If the authentication function unit of ACSE (see ISO 8649/Amd.1 and ISO 8650/Amd.1) is used in establishing the association, the value of the attribute Authentication Employed shall be true and the attribute Authentication Value shall be present.

The attributes AP Title of MMS Client, AE Qualifier of MMS Client, AP Invocation-identifier of MMS Client, AE Invocation-identifier of MMS Client, and Authentication Value, if present, shall be derived from the respective fields in the AARQ-apdu or AARE-apdu.

If the MMS user receives the MMS Initiate Request, the fields are calling-AP-title, calling-AE-qualifier, calling-AP-invocation-id, calling-AE-invocation-id, and calling-authentication-value (if present) in the AARQ-apdu.

If the MMS user is sending the MMS Initiate Request, the fields are responding-AP-title, responding-AE-qualifier, responding-AP-invocation-id, responding-AE-invocation-id, and responding-authentication-value (if present) in the AARE-apdu.

In 17.8, third paragraph, change:

clauses 6 to 16, 20, and

to:

clauses 6 to 16, 20, 21, and

In 17.11 on the last line, change:

shall be 2.

to:

shall be three.

Renumber 17.2 through 17.12 to be 17.3 through 17.13.

18. Conformance

In 18.3, table 2, page 106, add to the end of the table:

DefineAccessControlList	
GetAccessControlListAttributes	
ReportAccessControlledObjects	
DeleteAccessControlList	
ChangeAccessControl	

In 18.4, table 3, page 107, add to the end of the table:

ACO	

19. MMS Abstract Syntax

(This amendment makes no changes to clause 19 of ISO/IEC 9506-2.)

20. Data Exchange Protocol

In 20.3 of amendment 1 to ISO 9506-2, change the second production to:

```
GetDataExchangeAttributes-Response ::=  SEQUENCE {
      inUse[0] IMPLICIT BOOLEAN,
      listOfRequestTypeSpecifications[1] IMPLICIT SEQUENCE OF
      TypeSpecification,
      listOfResponseTypeSpecifications[2] IMPLICIT SEQUENCE OF
      TypeSpecification,
      programInvocation[3] IMPLICIT Identifier OPTIONAL,
      accessControlList[4] IMPLICIT Identifier OPTIONAL
                  -- Shall not appear in minor version one or two
       }
```

Add a new subclause 20.3.2.2:

20.3.2.2 Access Control List

The accessControlList parameter shall appear if and only if the ACO-CBB has been negotiated.

Add a new clause 21:

21. Conditioned Service Response Protocol

21.1 Introduction

This clause describes the protocol required for the realization of the services that are defined in IS0/IEC 9506-1, clause 21. This includes:

DefineAccessControlList
GetAccessControlListAttributes
ReportAccessControlledObjects
DeleteAccessControlList
ChangeAccessControl

21.2 AccessCondition

The abstract syntax of the Access Condition parameter is specified by the AccessCondition type. This type is specified below and described in the paragraphs that follow. Subclause 5.5 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
    AccessCondition ::= CHOICE {
         never          [0] IMPLICIT NULL,
         semaphore      [1] ObjectName, -- Semaphore Name
         user           [2] CHOICE {
associationApplicationReference,
noneNULL
},
         password       [3] Authentication-value,
         joint          [4] IMPLICIT SEQUENCE OF AccessCondition,
         alternate      [5] IMPLICIT SEQUENCE OF AccessCondition
         }

   -- Authentication-value is defined in ISO 8650/Amd.1
```

21.3 DefineAccessControlList

The abstract syntax of the defineAccessControlList choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified by the DefineAccessControlList-Request and DefineAccessControlList-Response types, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 describes the derivation of all parameters for which explicit derivations are not provided in this Subclause.

```
  DefineAccessControlList-Request ::= SEQUENCE {
   accessControlListName      [0] IMPLICIT Identifier,
   accessControlListElements [1] IMPLICIT SEQUENCE {
      readAccessCondition        [0] AccessCondition OPTIONAL,
      storeAccessCondition       [1] AccessCondition OPTIONAL,
      writeAccessCondition       [2] AccessCondition OPTIONAL,
      loadAccessCondition        [3] AccessCondition OPTIONAL,
      executeAccessCondition     [4] AccessCondition OPTIONAL,
      deleteAccessCondition      [5] AccessCondition OPTIONAL,
      editAccessCondition        [6] AccessCondition OPTIONAL
       }
  }


   DefineAccessControlList-Response ::= NULL
```

21.3.1 DefineAccessControlList-Request

The abstract syntax of the defineAccessControlList-Request choice of the Confirmed-ServiceRequest type shall be the DefineAccessControlList-Request.

21.3.1.1 accessControlListElements

The accessControlListElements field shall be the List of Access Control Element parameter of the DefineAccessControlList.request primitive and shall appear as the List of Access Control Element parameter of the DefineAccessControlList.indication. This field shall contain zero or one occurrence of the AccessCondition type for each of the seven possible values of the Service Class parameter. The AccessCondition associated with the Service Class equal to Read shall appear in the readAccessCondition field of the accessControlListElement; similarly for the other values of the Service Class parameter.

21.3.2 DefineAccessControlList-Response

The abstract syntax of the defineAccessControlList-Response choice of the Confirmed-ServiceResponse type shall be the DefineAccessControlList-Response.

21.4 GetAccessControlListAttributes

The abstract syntax of the getAccessControlListAttributes choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified by the GetAccessControlListAttributes-Request and GetAccessControlListAttributes-Response types, respectively. These types are specified below and described in the paragraphs that follow. Clause describes the derivation of all parameters for which explicit derivations are not provided in this Subclause.

```
GetAccessControlListAttributes-Request ::= CHOICE {
     accessControlListName  [0] IMPLICIT Identifier,
     vMD                    [1] IMPLICIT NULL,
     namedObject            [2] IMPLICIT SEQUENCE {
      extendedObjectClass   [0] CHOICE {
        objectClass             [0] IMPLICIT INTEGER {
             namedVariable          (0),
             scatteredAccess        (1),
             namedVariableList      (2),
             namedType              (3),
             semaphore              (4),
             eventCondition         (5),
             eventAction            (6),
             eventEnrollment        (7),
             journal                (8),
             domain                 (9),
             programInvocation     (10),
             operatorStation       (11),
             dataExchange          (12),
             accessControlList     (13)
             } },
       objectName               [1] ObjectName
       }
  }
```

```
GetAccessControlListAttributes-Response ::= SEQUENCE {
  name                [0] Identifier, -- Access Control List name
  accessControlListElements       [1] IMPLICIT SEQUENCE {
    readAccessCondition     [0] AccessCondition OPTIONAL,
    storeAccessCondition    [1] AccessCondition OPTIONAL,
    writeAccessCondition    [2] AccessCondition OPTIONAL,
    loadAccessCondition     [3] AccessCondition OPTIONAL,
    executeAccessCondition [4] AccessCondition OPTIONAL,
    deleteAccessCondition   [5] AccessCondition OPTIONAL,
    editAccessCondition     [6] AccessCondition OPTIONAL
    },
  vMDuse              [2] IMPLICIT BOOLEAN,
  references          [3] IMPLICIT SEQUENCE OF SEQUENCE {
      extendedObjectClass    [0] CHOICE {
        objectClass                [0] IMPLICIT INTEGER {
          namedVariable            (0),
          scatteredAccess          (1),
          namedVariableList        (2),
          namedType                (3),
          semaphore                (4),
          eventCondition           (5),
          eventAction              (6),
          eventEnrollment          (7),
          journal                  (8),
          domain                   (9),
          programInvocation       (10),
          operatorStation         (11),
          dataExchange            (12),
          accessControlList       (13)
          } },
     objectCount          [1] IMPLICIT INTEGER
     },
   accessControlList           [4] IMPLICIT Identifier OPTIONAL
                -- shall be included if and only if
                -- ACO has been negotiated
}
```

21.4.1 GetAccessControlListAttributes-Request

The abstract syntax of the getAccessControlListAttributes-Request choice of the Confirmed-ServiceRequest type shall be the GetAccessControlListAttributes-Request.

21.4.2 GetAccessControlListAttributes-Response

The abstract syntax of the getAccessControlListAttributes-Response choice of the Confirmed-ServiceResponse type shall be the GetAccessControlListAttributes-Response.

21.4.2.1 accessControlListElements

The accessControlListElements field shall be the List of Access Control Element parameter of the DefineAccessControlList.response primitive and shall appear as the List of Access Control Element parameter of the DefineAccessControlList.confirm. This field shall contain zero or one occurrence of the AccessCondition type for each of the seven possible values of the Service Class parameter. The AccessCondition associated with the Service Class equal to Read shall appear in the readAccessCondition field of the accessControlListElement; similarly for the other values of the Service Class parameter.

21.4.2.2 Counts of Controlled Objects

The abstract syntax of the Counts of Controlled Objects parameter shall be the references field of the GetAccessControlListAttributes-Response type. For each object class that contains one or more objects that reference this Access Control List object, a sequence of objectClass and objectCount shall be included. If there are no objects in a class that reference this Access Control List object (that is, the count is zero), this sequence shall not be included. If ACO has not been negotiated, this field shall not appear.

21.4.2.3 Access Control List

The accessControlList parameter shall appear if and only if the ACO CBB has been negotiated.

21.5 ReportAccessControlledObjects

The abstract syntax of the reportAccessControlledObjects choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified by the ReportAccessControlledObjects-Request and ReportAccessControlledObjects-Response types, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 describes the derivation of all parameters for which explicit derivations are not provided in this clause.