INTERNATIONAL STANDARD ISO/IEC 9506-2:1990
TECHNICAL CORRIGENDUM 1

Published 1995-12-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ELECTROTECHNIQUE INTERNATIONALE

Industrial automation systems — Manufacturing Message Specification —

Part 2:
Protocol specification

TECHNICAL CORRIGENDUM 1

Systèmes d'automatisation industrielle — Spécification de messagerie industrielle —

Partie 2: Spécification de protocole

RECTIFICATIF TECHNIQUE 1

Technical corrigendum 1 to International Standard ISO/IEC 9506-2:1990 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture and communications.*

Page 12

Subclause 5.14

In the note, lines 1 and 2, change

"this part of this part of"

to

"this part of"

Page 16

Subclause 6.3.1.2

In item a) in the third line change "error class SERVICE" to "error class SERVICE-PREEMPT".

ICS 25.040.40 **Ref. No. ISO/IEC 9506-2:1990/Cor.1:1995(E)**

Descriptors: automation, automation engineering, manufacturing, messages, computer applications, data processing, information interchange, network interconnection, open systems interconnection, communication procedure, protocols.

Printed in Switzerland

In item b) in the second line change

"error class SERVICE" to "error class SERVICE-PREEMPT".

Page 19

Clause 7

Change

```
{ iso standard 8650 abstract-syntax(2) acse-pdi(1) }
```

to

```
{ joint-iso-ccitt association-control(2) abstract(1) }
```

Page 19

Subclause 7.1

Replace the comment line in the Confirmed-RequestPDU production. The protocol in this subclause should now read:

```
Confirmed-RequestPDU ::= SEQUENCE {
  invokeID              Unsigned32,
  listOfModifier        SEQUENCE OF Modifier OPTIONAL,
  ConfirmedServiceRequest,
  [79] CS-Request-Detail   OPTIONAL,
    -- shall not be transmitted if value is the value
    -- of a tagged type derived from NULL
  }
```

Page 20

Subclause 7.1

At the end of the subclause, that is, immediately preceding 7.2, add the following note:

NOTE — The intent of the comment in the Confirmed-RequestPDU production is to preclude the transmission of the CS-Request-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Request-Detail type. (See the type definitions in 7.5.2 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 20

Subclause 7.2

Replace the comment line in the Unconfirmed-PDU production.
The protocol in this subclause should now read:

```
Unconfirmed-PDU ::= SEQUENCE {
  UnconfirmedService,
  [79] CS-Unconfirmed-Detail    OPTIONAL,
    -- shall not be transmitted if value is the value
    -- of a tagged type derived from NULL
  }
```

Page 20

Subclause 7.2

At the end of the subclause, that is immediately preceding 7.3, add the following note:

NOTE — The intent of the comment in the Unconfirmed-PDU production is to preclude the transmission of the CS-Unconfirmed-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Unconfirmed-Detail type. (See the type definitions in 7.5.3 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 20

Subclause 7.3

Replace the comment line in the Confirmed-ResponsePDU production.
The protocol in this subclause should now read:

```
Confirmed-ResponsePDU ::= SEQUENCE {
  invokeID              Unsigned32,
  ConfirmedServiceResponse,
  [79] CS-Response-Detail  OPTIONAL,
    -- shall not be transmitted if value is the value
    -- of a tagged type derived from NULL
  }
```

Page 20

Subclause 7.3

At the end of the subclause, that is immediately preceding 7.4, add the following note:

NOTE — The intent of the comment in the Confirmed-ResponsePDU production is to preclude the transmission

of the CS-Response-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Response-Detail type. (See the type definitions in 7.5.4 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 32

Subclause 7.6.2

In the definition of Identifier, change "__" to "_" (only a single underscore)

Page 36

Subclause 8.2

Remove 'akec' named bit from the ParameterSupportOptions production.
This production should now read:

```
ParameterSupportOptions  ::= BIT STRING {
 str1    (0),
 str2    (1),
 vnam    (2),
 valt    (3),
 vadr    (4),
 vsca    (5),
 tpy     (6),
 vlis    (7),
 real    (8),
 --          bit 9 reserved for future definition
 cei     (10)
 }
```

Page 47

Subclause 10.8

Replace the line

```
listOfCapabilities  [1] IMPLICIT SEQUENCE OF VisibleString,
```

with

```
listOfCapabilities  [1] IMPLICIT SEQUENCE OF VisibleString OPTIONAL,
```

Page 48

Subclause 10.8.1

Add the following text at the end of the present text:

If the List Of Capabilities parameter is present in the service request, and the parameter specifies an empty list, a SEQUENCE OF VisibleString with zero elements shall be transmitted; if the parameter is not present in the service request, this field shall not be transmitted.

Page 48

Subclause 10.10

Replace the line

```
listOfCapabilities    [1] IMPLICIT SEQUENCE OF VisibleString,
```

with

```
listOfCapabilities    [1] IMPLICIT SEQUENCE OF VisibleString OPTIONAL,
```

Page 48

Subclause 10.10.1

Add the following text at the end of the present text:

If the List Of Capabilities parameter is present in the service request, and the parameter specifies an empty list, a SEQUENCE OF VisibleString with zero elements shall be transmitted; if the parameter is not present in the service request, this field shall not be transmitted.

Page 59

Subclause 12.4.2

Add an ASN.1 comment following the unsigned choice. That line should now read:

```
unsigned    [6] IMPLICIT INTEGER, -- shall not be negative
```

Add an ASN.1 comment following the bcd choice. That line should now read:

```
bcd         [13] IMPLICIT INTEGER, -- shall not be negative
```

Page 62

Subclause 12.4.3

Replace the last line of the protocol with the following:

```
    object-non-existent    (10), -- OBJECT-NON-EXISTENT
    object-value-invalid   (11)  -- OBJECT-VALUE-INVALID
```

Page 80

Subclause 15.8

Replace the comment line in the DefineEventAction-Request production. The protocol in this subclause should now read:

```
DefineEventAction-Request ::= SEQUENCE {
   eventActionname           [0] ObjectName,
   listOfModifier            [1]  IMPLICIT SEQUENCE  OF Modifier OPTIONAL,
   confirmedServiceRequest   [2] ConfirmedServiceRequest,
   cs-extension              [79] CS-Request-Detail  OPTIONAL,
     -- shall not be transmitted if value is the value
     -- of a tagged type derived from NULL
   }
```

Page 80

Subclause 15.8.1

At the end of the subclause add the following note:

NOTE — The intent of the comment in the DefineEventAction-Request production is to preclude the transmission of the CS-Request-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Request-Detail type. (See the type definitions in 7.5.2 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 81

Subclause 15.10

Replace the comment line in the GetEventActionAttributes-Response production. The protocol in this subclause should now read:

```
GetEventActionAttributes-Response ::= SEQUENCE {
    mmsDeletable              [0] IMPLICIT BOOLEAN DEFAULT FALSE,
    listOfModifier            [1] IMPLICIT SEQUENCE OF Modifier,
    confirmedServiceRequest   [2] ConfirmedServiceRequest,
    cs-extension              [79] CS-Request-Detail OPTIONAL,
    -- shall not be transmitted if value is the value
    -- of a tagged type derived from NULL
    }
```

Page 81

Subclause 15.10.2

At the end of the subclause add the following explanatory note:

> NOTE — The intent of the comment in the GetEventActionAttributes-Response production is to preclude the transmission of the CS-Response-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Response-Detail type. (See the type definitions in 7.5.2 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 82

Subclause 15.12

Delete the last line of the DefineEventEnrollment-Request.
The production should read:

```
DefineEventEnrollment-Request ::= SEQUENCE {
  eventEnrollmentName        [0] ObjectName,
  eventConditionName         [1] ObjectName,
  eventConditionTransitions  [2] IMPLICIT Transitions,
  alarmAcknowledmentRule     [3] IMPLICIT AlarmAckRule,
  eventActionName            [4] ObjectName OPTIONAL,
  clientApplication          [5] ApplicationReference OPTIONAL
  }
```

Page 84

Subclause 15.14

Replace the comment line following the item tagged [9] in the EventEnrollment production with:

```
-- shall not be transmitted if the value is NULL
```

Delete the acknowledgementEventCondition element from the sequence.

The protocol in this subclause should now read:

```
EventEnrollment  ::= SEQUENCE {
  eventEnrollmentName            [0] ObjectName,
  eventConditionName             [1] CHOICE {
    eventCondition                   [0] ObjectName,
    undefined                        [1] IMPLICIT NULL
    },
  eventActionName                [2] CHOICE {
    eventAction                      [0] ObjectName,
    undefined                        [1] IMPLICIT NULL
    }  OPTIONAL,
```

```
    clientApplication               [3] ApplicationReference OPTIONAL,
    mmsDeletable                    [4] IMPLICIT BOOLEAN DEFAULT FALSE,
    enrollmentClass                 [5] IMPLICIT EE-Class,
    duration                        [6] IMPLICIT EE-Duration DEFAULT current,
    invokeID                        [7] IMPLICIT Unsigned32 OPTIONAL,
    remainingAcceptableDelay        [8] IMPLICIT Unsigned32 OPTIONAL,
    additionalDetail                [9] IMPLICIT EE-Additional-Detail
                                            OPTIONAL
      -- shall not be transmitted if the value is NULL
    }
```

Page 85

Subclause 15.14.2.1.4

At the end of the subclause add the following note:

> NOTE — The intent of the comment in the EventEnrollment production is to preclude the transmission of the EE-Additional-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to the EE-Additional-Detail type. (See the type definition in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the EE-Additional-Detail type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 85

Subclause 15.14.2.1.5

Delete all of the subclause.

Page 87

Subclause 15.17

Replace the comment line in the EventNotification production.
The protocol in this subclause should now read:

```
EventNotification  ::= SEQUENCE {
   eventEnrollmentName [0] ObjectName,
   eventConditionName  [1] ObjectName,
   severity            [2] IMPLICIT Severity,
   currentState        [3] IMPLICIT EC-State OPTIONAL,
   transitionTime      [4] EventTime,
   notificationLost    [6] IMPLICIT BOOLEAN DEFAULT FALSE,
   alarmAcknowledgmentRule  [7] IMPLICIT AlarmAckRule OPTIONAL,
   actionResult        [8] IMPLICIT SEQUENCE {
     eventActionName    ObjectName,
     eventActionResul     CHOICE {
      success               [0] IMPLICIT SEQUENCE {
                        ConfirmedServiceResponse,
                        [79] CS-Response-Detail OPTIONAL,
                        -- shall not be transmitted if value is the
```

```
                    -- value of a tagged type derived from NULL
                },
    failure                 [1] IMPLICIT SEQUENCE {
      modifierPosition          [0] IMPLICIT Unsigned32 OPTIONAL,
      serviceError              [1] IMPLICIT ServiceError
      }
    }
  } OPTIONAL
}
```

Page 87

Subclause 15.17.1.1.1

At the end of the subclause add the following note:

> NOTE — The intent of the comment in the EventNotification production is to preclude the transmission of the CS-Response-Detail field for all occurrences in the abstract syntax defined in this part of ISO/IEC 9506. This is done by assigning the NULL type to all the types referenced in the CS-Response-Detail type. (See the type definitions in 7.5.4 and in clause 19.) MMS Companion Standards may also preclude the transmission of this field either by defining the appropriate type to be NULL, similarly to the definitions in clause 19, or by selecting a NULL choice from a CHOICE type.

Page 88

Subclause 15.18

Delete the last line of the AcknowledgeEventNotification-Request.
The production should now read:

```
AcknowledgeEventNotification-Request ::= SEQUENCE {
  eventEnrollmentName          [0] ObjectName,
  acknowledgedState            [2] IMPLICIT EC-State,
  timeOfAcknowledgedTransition [3] EventTime
  }
```

Page 89

Subclause 15.19

Replace the comment line following the item tagged [4] in the AlarmSummary production. The protocol in this subclause should now read:

```
AlarmSummary ::= SEQUENCE {
    eventConditionName  [0] ObjectName,
    severity            [1] IMPLICIT Unsigned8,
    currentState        [2] IMPLICIT EC-State,
    unacknowledgedState [3] IMPLICIT INTEGER {
       none      (0),       -- NONE
       active    (1),       -- ACTIVE
       idle      (2),       -- IDLE
```