TECHNICAL REPORT

ISO/IEC TR 13345

First edition
1994-07-15

Industrial automation systems — Specification of subsets of the protocol for ISO/IEC 9506

Systèmes d'automatisation industrielle — Spécification de sous-ensembles du protocole pour l'ISO/CEI 9506

Reference number
ISO/IEC TR 13345:1994(E)

Contents

Page

Foreword ... viii

1 Scope ... 1

2 Normative references ... 1

3 Protocol subset specification ... 2
3.1 Notation ... 2
3.2 Determination of the effective protocol ... 3
3.3 Protocol for companion standards ... 4

4 Protocol module ... 6
4.1 MMSpdu ... 6
4.2 Confirmed-RequestPDU ... 7
4.3 Unconfirmed-PDU ... 7
4.4 Confirmed-ResponsePDU ... 7
4.5 Confirmed-ErrorPDU ... 8
4.6 Modifier ... 8
4.7 ConfirmedServiceRequest ... 8
4.8 AdditionalService-Request ... 14
4.9 CS-Request-Detail ... 16
4.10 UnconfirmedService ... 16
4.11 CS-Unconfirmed-Detail ... 17
4.12 ConfirmedServiceResponse ... 17
4.13 AdditionalService-Response ... 23
4.14 CS-Response-Detail ... 25
4.15 ServiceError ... 25
4.16 AdditionalService-Error ... 28
4.17 TimeOfDay ... 29
4.18 Identifier ... 29
4.19 Integer's ... 29
4.20 ObjectName ... 29
4.21 ApplicationReference ... 30
4.22 FileName ... 30
4.23 Priority ... 30
4.24 Severity ... 30
4.25 Initiate-PDU's ... 30
4.26 Initiate-Detail's ... 31
4.27 ServiceSupportOptions ... 31
4.28 ParameterSupportOptions ... 34

4.29 Conclude-PDU's . . . 34
4.30 Cancel-PDS's . . . 35
4.31 RejectPDU . . . 35
4.32 Status . . . 36
4.33 Robot Status . . . 37
4.34 UnsolicitedStatus . . . 37
4.35 GetNameList . . . 38
4.36 CsAdditionalObjectClasses . . . 39
4.37 Identify . . . 39
4.38 Rename . . . 39
4.39 GetCapabilityList . . . 40
4.40 InitiateDownloadSequence . . . 40
4.41 DownloadSegment . . . 40
4.42 TerminateDownloadSequence . . . 40
4.43 InitiateUploadSequence . . . 41
4.44 UploadSegment . . . 41
4.45 TerminateUploadSequence . . . 41
4.46 RequestDomainDownload . . . 41
4.47 RequestDomainUpload . . . 41
4.48 LoadDomainContent . . . 42
4.49 StoreDomainContent . . . 42
4.50 DeleteDomain . . . 42
4.51 GetDomainAttributes . . . 42
4.52 DomainState . . . 43
4.53 CreateProgramInvocation . . . 43
4.54 DeleteProgramInvocation . . . 44
4.55 Start . . . 44
4.56 ProgramInvocationState . . . 44
4.57 Stop . . . 45
4.58 Resume . . . 45
4.59 Reset . . . 45
4.60 Kill . . . 45
4.61 GetProgramInvocationAttributes . . . 46
4.62 TypeSpecification . . . 46
4.63 AlternateAccess . . . 47
4.64 AccessResult . . . 48
4.65 Data . . . 48
4.66 DataAccessError . . . 49
4.67 VariableAccessSpecification . . . 49
4.68 VariableSpecification . . . 49
4.69 ScatteredAccessDescription . . . 50
4.70 Address . . . 50
4.71 Read . . . 50
4.72 Write . . . 50
4.73 InformationReport . . . 50
4.74 GetVariableAccessAttributes . . . 51

4.75 DefineNamedVariable . . . 51
4.76 DefineScatteredAccess . . . 51
4.77 GetScatteredAccessAttributes . . . 51
4.78 DeleteVariableAccess . . . 52
4.79 DefineNamedVariableList . . . 52
4.80 GetNamedVariableListAttributes . . . 52
4.81 DeleteNamedVariableList . . . 53
4.82 DefineNamedType . . . 53
4.83 GetNamedTypeAttributes . . . 53
4.84 DeleteNamedType . . . 53
4.85 TakeControl . . . 54
4.86 RelinquishControl . . . 54
4.87 DefineSemaphore . . . 54
4.88 DeleteSemaphore . . . 55
4.89 ReportSemaphoreStatus . . . 55
4.90 ReportPoolSemaphoreStatus . . . 55
4.91 ReportSemaphoreEntryStatus . . . 55
4.92 AttachToSemaphore . . . 56
4.93 SemaphoreEntry . . . 56
4.94 Input . . . 56
4.95 Output . . . 56
4.96 DefineEventCondition . . . 57
4.97 DeleteEventCondition . . . 57
4.98 GetEventConditionAttributes . . . 57
4.99 ReportEventConditionStatus . . . 58
4.100 AlterEventConditionMonitoring . . . 58
4.101 TriggerEvent . . . 59
4.102 DefineEventAction . . . 59
4.103 DeleteEventAction . . . 59
4.104 GetEventActionAttributes . . . 59
4.105 ReportEventActionStatus . . . 60
4.106 DefineEventEnrollment . . . 60
4.107 DeleteEventEnrollment . . . 61
4.108 GetEventEnrollmentAttributes . . . 61
4.109 EventEnrollment . . . 61
4.110 ReportEventEnrollmentStatus . . . 62
4.111 AlterEventEnrollment . . . 62
4.112 Eventnotification . . . 62
4.113 AcknowledgeEventNotification . . . 63
4.114 GetAlarmSummary . . . 63
4.115 GetAlarmEnrollmentSummary . . . 64
4.116 AttachToEventCondition . . . 65
4.117 EC-Class . . . 65
4.118 EC-State . . . 65
4.119 EE-State . . . 65
4.120 Transitions . . . 66

4.121 AlarmAckRule . . . 66
4.122 EE-Class . . . 66
4.123 EE-Duration . . . 66
4.124 EN-Additional-Detail . . . 66
4.125 EventTime . . . 67
4.126 ReadJournal . . . 67
4.127 WriteJournal . . . 67
4.128 InitializeJournal . . . 67
4.129 ReportJournalStatus . . . 68
4.130 CreateJournal . . . 68
4.131 DeleteJournal . . . 68
4.132 EntryContent . . . 68
4.133 ExchangeData . . . 69
4.134 GetDataExchangeAttributes . . . 69
4.135 AccessCondition . . . 69
4.136 DefineAccessControlList . . . 69
4.137 GetAccesscontrolListAttributes . . . 70
4.138 ReportAccessControlledObjects . . . 71
4.139 DeleteAccessControlList . . . 71
4.140 ChangeAccessControl . . . 72
4.141 ObtainFile . . . 73
4.142 VMDStop . . . 73
4.143 VMDReset . . . 73
4.144 Select . . . 73
4.145 AlterProgramInvocationAttributes . . . 73
4.146 ControlElement . . . 74
4.147 InitiateUnitControlLoad . . . 74
4.148 UnitControlLoadSegment . . . 74
4.149 UnitControlUpload . . . 74
4.150 StartUnitControl . . . 75
4.151 StopUnitControl . . . 75
4.152 CreateUnitControl . . . 75
4.153 AddToUnitControl . . . 75
4.154 RemoveFromUnitControl . . . 76
4.155 GetUnitControlAttributes . . . 76
4.156 LoadUnitControlFromFile . . . 76
4.157 StoreUnitControlToFile . . . 76
4.158 DeleteUnitControl . . . 77
4.159 DefineEventConditionList . . . 77
4.160 DeleteEventConditionList . . . 77
4.161 AddEventConditionListReference . . . 77
4.162 RemoveEventConditionListReference . . . 78
4.163 GetEventConditionListAttributes . . . 78
4.164 ReportEventConditionListStatus . . . 78
4.165 EventConditionStatus . . . 79
4.166 AlterEventConditionListMonitoring . . . 79

Annex A File Management Protocol 80
A.1 FileOpen 80
A.2 FileRead 80
A.3 FileClose 80
A.4 FileRename 81
A.5 FileDelete 81
A.6 FileDirectory 81

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 13345, which is a Technical Report of type 2, was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC 5, *Architecture and communication.*

This document is being issued in the type 2 Technical Report series of publications (according to subclause G.4.2.2 of part 1 of the ISO/IEC Directives, 1992) as a "prospective standard for provisional application" in the field of industrial automation because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not be regarded as an "International Standard". It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO Central Secretariat.

A review of this type 2 Technical Report will be carried out not later than two years after its publication with the options of: extension for another two years; conversion into an International Standard; or withdrawal.

Annex A forms an integral part of this Technical Report.

Industrial automation systems — Specification of subsets of the protocol for ISO/IEC 9506

1 Scope

ISO/IEC 9506-1 and ISO/IEC 9506-2 were published in 1990. Since publication, considerable experience has been gained by implementors identifying defects and other shortcomings. One major shortcoming of ISO/IEC 9506-2 is that although it allows subsetting of the protocol in accordance with parameters declared at the time of association establishment, no clear notation for describing the subsets is present. Moreover, for some combinations of parameters, there are possible ambiguities in the specification of the subset.

MMS prescribes a procedure, used at the time of association establishment, in which sets of parameters are exchanged for the purpose of identifying the services that may be performed during the association. The effect of identifying these parameters is such that the protocol available to be used during the association is limited to a proper subset of the entire protocol specified in ISO/IEC 9506-2. The parameters exchanged are of two types:

a) those that are announced by the two MMS users, the service conformance building block (CBB),

b) those that are negotiated between the two MMS users, the parameter CBB. These CBBs are proposed by the association initiator and either accepted or rejected by the association responder. Negotiation always works to reduce the set proposed, never to augment it.

Declaration of support of any service CBB requires inclusion of the protocol related to that service in the protocol set to be used on that association. Support of a parameter CBB usually results in the inclusion of some optional fields within the protocol of some service request or response. However, in some cases, support of a parameter CBB implies support of one or more service CBBs, regardless of whether or not support for those service CBBs has been declared. It is in such situations that ambiguity may arise. This Technical Report specifies a uniform technique for resolving such ambiguities.

It introduces a formal notation to describe the subsets in an unambiguous form, and a complete specification of the MMS protocol using this new notation. The notation also supports the protocol of the Companion Standards. The protocol of the Robot Companion Standard (ISO/IEC 9506-3), of the NCC Companion Standard (ISO/IEC 9506-4), and of the Process Industries Companion Standard (the subject of future International Standard ISO/IEC 9506-6) are included.

Annex A specifies the File Management Protocol, the subject of annex C of ISO/IEC 9506-2.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this Technical Report. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Technical Report are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9506-1:1990, *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition.*

ISO/IEC 9506-2:1990, *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification.*

ISO/IEC 9506-1/Amd.1:1993, *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition AMENDMENT 1 Data exchange.*

ISO/IEC 9506-2/Amd.1:1993, *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification AMENDMENT 1 Data exchange.*

ISO/IEC 9506-1/Amd.2:—[1)], *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition AMENDMENT 2 Conditioned service response.*

ISO/IEC 9506-2/Amd.2:—[1)], *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification AMENDMENT 2 Conditioned service response.*

ISO/IEC 9506-3:1991, *Industrial automation systems — Manufacturing Message Specification — Part 3: Companion standard for robotics.*

ISO/IEC 9506-4:1992, *Industrial automation systems — Manufacturing Message Specification — Part 4: Companion standard for numerical control.*

ISO/IEC 8824:1990, *Information technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).*

3 Protocol subset specification

3.1 Notation

The notation being introduced by this technical report has the form of a preprocessor language in which ASN.1 is embedded. It is very similar in concept to the macro preprocessor for the C language. There are only three commands used in this notation:

- IF (<list of arguments>)

- ELSE

- ENDIF.

1) To be published.

The IF command requires an argument list enclosed in parentheses; the arguments are the names of the conformance building blocks, either service or parameter. One or more such arguments must appear. If there is more than one argument, the arguments are separated by one or more spaces. The argument is treated as a boolean variable that has the value true if the corresponding service or parameter building block is supported as a result of the MMS Initiate exchange. If there is one argument, then the lines following the IF statement up to the ELSE statement or to the matching ENDIF statement (if no ELSE statement appears) are to be included in the resulting ASN.1 definition if and only if the conformance building block so named in supported. If there is more than one argument, the lines following the IF statement are to be included if any of the conformance building blocks in the argument list is supported. (This can be thought of as a 'logical OR' function of the conformance building blocks.)

IF statements may be nested to any depth; the effect of

```
IF ( x )
IF ( y )
```

is to include the lines following these commands if and only if both x and y are true, that is if conformance block x and conformance block y are both included. (This can be thought of as a 'logical AND' function of the conformance building blocks.)

The ELSE statement may appear to allow ASN.1 statements to be included if a conformance building block is not true. Its use is similar to the normal use of ELSE in programming languages.

The ENDIF statement is used to end the scope of an IF statement or ELSE statement. Each IF statement must have a matching ENDIF statement.

3.2 Determination of the effective protocol

The protocol effective for any specific combination of service and parameter CBBs can be determined by the following procedure:

a) For each service CBB and parameter CBB declared or negotiated by the Initiate exchange, set the corresponding argument equal to true.

b) Process the entire ASN.1 module specified in clause 4 and in Annex A. For each IF statement, evaluate the argument.

i) If any of the elements in the argument is true, retain the statements following the IF statement up to a matching ENDIF or ELSE statement, if present. Discard the statements following the ELSE statement up to the matching ENDIF.

ii) If all the elements of the argument are false, discard the statements following up to the matching ELSE or ENDIF. If an ELSE statement is present, retain the statements following it to the matching ENDIF.

iii) Discard the IF statement, its matching ENDIF, and the ELSE, if present. The result should be an ASN.1 module devoid of IF, ELSE, and ENDIF statements.

c) In each production replace any occurrence of a comma followed immediately by a right brace with a right brace; i.e., delete such commas.

d) Form an ASN.1 working module of productions containing only the first production (i.e. the production MMSPDU from 4.1).

e) Add to the ASN.1 working module any productions referenced in the working module that are not already contained in that module.

f) Repeat step e) until no new productions are added.

The resulting ASN.1 module is the module that is effective for this combination of CBBs. Receipt of a PDU which is not consistent with this module should result in a reject.

3.3 Protocol for companion standards

In addition to the protocol of ISO 9506-2, the protocol for all the amendments and companion standards has also been included in this report. In order to generate the protocol for a companion standard, the set of parameter CBBs has been augmented with three new symbols:

csr - true if this is an instance of the companion standard for robots,

csnc - true if this is an instance of the companion standard for NC equipment,

cspi - true if this is an instance of the companion standard for the process industries.

Except for the fact that the companion standards define a different abstract syntax that must be negotiated using presentation negotiation, each companion standard acts just like a parameter CBB.

In order to accomodate these companion standards, the list of service and parameter CBBs is augmented by the service and parameter CBBs from those standards.

The complete list of service CBBs is:

status
getNameList
identify
rename
read
write
getVariableAccessAttributes
defineNamedVariable
defineScatteredAccess
getScatteredAccessAttributes
deleteVariableAccess
defineNamedVariableList
getNamedVariableListAttributes
deleteNamedVariableList
defineNamedType
getNamedTypeAttributes
deleteNamedType
input
output
takeControl
relinquishControl
defineSemaphore

deleteSemaphore
reportSemaphoreStatus
reportPoolSemaphoreStatus
reportSemaphoreEntryStatus
initiateDownloadSequence
downloadSegment
terminateDownloadSequence
initiateUploadSequence
uploadSegment
terminateUploadSequence
requestDomainDownload
requestDomainUpload
loadDomainContent
storeDomainContent
deleteDomain
getDomainAttributes
createProgramInvocation
deleteProgramInvocation
start
stop
resume
reset
kill
getProgramInvocationAttributes
obtainFile
defineEventCondition
deleteEventCondition
getEventConditionAttributes
reportEventConditionStatus
alterEventConditionMonitoring
triggerEvent
defineEventAction
deleteEventAction
getEventActionAttributes
reportEventActionStatus
defineEventEnrollment
deleteEventEnrollment
alterEventEnrollment
reportEventEnrollmentStatus
getEventEnrollmentAttributes
acknowledgeEventNotification
getAlarmSummary
getAlarmEnrollmentSummary
readJournal
writeJournal
initializeJournal
reportJournalStatus
createJournal
deleteJournal
getCapabilityList
fileOpen
fileRead
fileClose
fileRename
fileDelete
fileDirectory
unsolicitedStatus
informationReport
eventNotification
attachToEventCondition
attachToSemaphore
conclude
cancel
getDataExchangeAttributes
exchangeData
vMDStop
vMDReset
select
alterProgramInvocationAttributes
initiateUnitControlLoad
unitControlLoadSegment
unitControlUpload
startUnitControl
stopUnitControl
createUnitControl
addToUnitControl
removeFromUnitControl
getUnitControlAttributes
loadUnitControlFromFile
storeUnitControlToFile
deleteUnitControl
defineEventConditionList
deleteEventConditionList
addEventConditionListReference
removeEventConditionListReference
getEventConditionListAttributes
reportEventConditionListStatus
alterEventConditionListMonitoring
defineAccessControlList
getAccessControlListAttributes
reportAccessControlledObjects
deleteAccessControlList
changeAccessControl

The complete list of parameter (and parameter-like) CBBs is:

str1
str2
vnam
valt
vadr
vsca
tpy
vlis
real
cei
des
dei
recl
csr
csnc
cspi

4 Protocol module

This clause presents the entire protocol module using the macro notation described in Clause 3. It includes all of the protocol for the companion standard for robots, ISO 9506-3, the companion standard for NC equipment, ISO 9506-4, and the companion standard for the process industries, ISO 9506-6. It also includes the protocol for the Data Exchange services and the Conditioned Service Response services.

4.1 MMSpdu

```
IF ( csr csnc cspi )
IF ( csr )
ISO-9506-MMS-ROBOT-1 {iso standard 9506 part(3) mms-robot-module-version1(2) }
ENDIF
IF ( csnc )
ISO-9506-MMS-NCCS-1 {iso standard 9506 part(4) mms-nccs-module-version1(2)}
ENDIF
IF ( cspi )
ISO-9506-MMS-PROCESS-1 {iso standard 9506 part(6) mms-process-module-version1(2)}
ENDIF
ELSE
ISO-9506-MMS-1 {iso standard 9506 part(2) mms-abstract-syntax-version1(1) }
ENDIF

DEFINITIONS ::= BEGIN

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)};

MMSpdu ::= CHOICE {
 confirmed-RequestPDU       [0] IMPLICIT Confirmed-RequestPDU,
 confirmed-ResponsePDU      [1] IMPLICIT Confirmed-ResponsePDU,
 confirmed-ErrorPDU         [2] IMPLICIT Confirmed-ErrorPDU,
IF ( unsolicitedStatus informationReport eventNotification )
 unconfirmed-PDU            [3] IMPLICIT Unconfirmed-PDU,
ENDIF
 rejectPDU                  [4] IMPLICIT RejectPDU,
IF (cancel)
 cancel-RequestPDU          [5] IMPLICIT Cancel-RequestPDU,
 cancel-ResponsePDU         [6] IMPLICIT Cancel-ResponsePDU,
 cancel-ErrorPDU            [7] IMPLICIT Cancel-ErrorPDU,
ENDIF
 initiate-RequestPDU        [8] IMPLICIT Initiate-RequestPDU,
 initiate-ResponsePDU       [9] IMPLICIT Initiate-ResponsePDU,
```

```
 initiate-ErrorPDU          [10] IMPLICIT Initiate-ErrorPDU,
 conclude-RequestPDU        [11] IMPLICIT Conclude-RequestPDU,
 conclude-ResponsePDU       [12] IMPLICIT Conclude-ResponsePDU,
 conclude-ErrorPDU          [13] IMPLICIT Conclude-ErrorPDU }
```

4.2 Confirmed-RequestPDU

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

4.3 Unconfirmed-PDU

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

4.4 Confirmed-ResponsePDU

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

4.5 Confirmed-ErrorPDU

```
Confirmed-ErrorPDU ::= SEQUENCE {
 invokeID                [0] IMPLICIT Unsigned32,
IF (attachToEventCondition attachToSemaphore )
 modifierPosition        [1] IMPLICIT Unsigned32 OPTIONAL,
ENDIF
 serviceError            [2] IMPLICIT ServiceError }
```

4.6 Modifier

```
Modifier ::= CHOICE {
IF ( attachToEventCondition )
 attachToEventCondition [0] IMPLICIT AttachToEventCondition,
ENDIF
IF ( attachToSemaphore )
 attachToSemaphore       [1] IMPLICIT AttachToSemaphore
ENDIF
 }
```

4.7 ConfirmedServiceRequest

```
ConfirmedServiceRequest ::= CHOICE {
IF ( status )
 status                     [0] IMPLICIT Status-Request,
ENDIF
IF ( getNameList )
 getNameList                [1] IMPLICIT GetNameList-Request,
ENDIF
IF ( identify )
 identify                   [2] IMPLICIT Identify-Request,
ENDIF
IF ( rename )
 rename                     [3] IMPLICIT Rename-Request,
ENDIF
IF ( read )
 read                       [4] IMPLICIT Read-Request,
ENDIF
IF ( write )
 write                      [5] IMPLICIT Write-Request,
ENDIF
IF ( vnam vadr )
IF ( getVariableAccessAttributes )
 getVariableAccessAttributes  [6] GetVariableAccessAttributes-Request,
ENDIF
ENDIF
IF ( vnam )
IF ( vadr )
```

```
IF ( defineNamedVariable )
 defineNamedVariable        [7] IMPLICIT DefineNamedVariable-Request,
ENDIF
ENDIF
ENDIF
IF ( vsca )
IF ( vnam vadr )
IF ( defineScatteredAccess )
 defineScatteredAccess      [8] IMPLICIT DefineScatteredAccess-Request,
ENDIF
IF ( getScatteredAccessAttributes )
 getScatteredAccessAttributes [9] GetScatteredAccessAttributes-Request,
ENDIF
ENDIF
ENDIF
IF ( vsca vnam )
IF ( deleteVariableAccess )
 deleteVariableAccess       [10] IMPLICIT DeleteVariableAccess-Request,
ENDIF
ENDIF
IF ( vlis )
IF ( vnam )
IF ( defineNamedVariableList )
 defineNamedVariableList    [11] IMPLICIT DefineNamedVariableList-Request,
ENDIF
IF ( getNamedVariableListAttributes )
 getNamedVariableListAttributes [12] GetNamedVariableListAttributes-Request,
ENDIF
IF ( deleteNamedVariableList )
 deleteNamedVariableList    [13] IMPLICIT DeleteNamedVariableList-Request,
ENDIF
ENDIF
ENDIF
IF ( vnam )
IF ( defineNamedType )
 defineNamedType            [14] IMPLICIT DefineNamedType-Request,
ENDIF
IF ( getNamedTypeAttributes )
 getNamedTypeAttributes     [15] GetNamedTypeAttributes-Request,
ENDIF
IF ( deleteNamedType )
 deleteNamedType            [16] IMPLICIT DeleteNamedType-Request,
ENDIF
ENDIF
IF ( input )
 input                      [17] IMPLICIT Input-Request,
ENDIF
IF ( output )
 output                     [18] IMPLICIT Output-Request,
```

```
ENDIF
IF ( takeControl )
 takeControl                [19] IMPLICIT TakeControl-Request,
ENDIF
IF ( relinquishControl )
 relinquishControl          [20] IMPLICIT RelinquishControl-Request,
ENDIF
IF ( defineSemaphore )
 defineSemaphore            [21] IMPLICIT DefineSemaphore-Request,
ENDIF
IF ( deleteSemaphore )
 deleteSemaphore            [22] DeleteSemaphore-Request,
ENDIF
IF ( reportSemaphoreStatus )
 reportSemaphoreStatus      [23] ReportSemaphoreStatus-Request,
ENDIF
IF ( reportPoolSemaphoreStatus )
 reportPoolSemaphoreStatus [24] IMPLICIT ReportPoolSemaphoreStatus-Request,
ENDIF
IF ( reportSemaphoreEntryStatus )
 reportSemaphoreEntryStatus    [25] IMPLICIT ReportSemaphoreEntryStatus-Request,
ENDIF
IF ( initiateDownloadSequence )
 initiateDownloadSequence  [26] IMPLICIT InitiateDownloadSequence-Request,
 downloadSegment           [27] IMPLICIT DownloadSegment-Request,
 terminateDownloadSequence [28] IMPLICIT TerminateDownloadSequence-Request,
ENDIF
IF ( initiateUploadSequence )
 initiateUploadSequence     [29] IMPLICIT InitiateUploadSequence-Request,
 uploadSegment              [30] IMPLICIT UploadSegment-Request,
 terminateUploadSequence    [31] IMPLICIT TerminateUploadSequence-Request,
ENDIF
IF ( requestDomainDownload )
 requestDomainDownload      [32] IMPLICIT RequestDomainDownload-Request.
ENDIF
IF ( requestDomainUpload )
 requestDomainUpload        [33] IMPLICIT RequestDomainUpload-Request,
ENDIF
IF ( loadDomainContent )
 loadDomainContent          [34] IMPLICIT LoadDomainContent-Request,
ENDIF
IF ( storeDomainContent )
 storeDomainContent         [35] IMPLICIT StoreDomainContent-Request,
ENDIF
IF ( deleteDomain )
 deleteDomain               [36] IMPLICIT DeleteDomain-Request,
ENDIF
IF ( getDomainAttributes )
 getDomainAttributes        [37] IMPLICIT GetDomainAttributes-Request,
```

```
ENDIF
IF ( createProgramInvocation )
 createProgramInvocation    [38] IMPLICIT CreateProgramInvocation-Request,
ENDIF
IF ( deleteProgramInvocation )
 deleteProgramInvocation    [39] IMPLICIT DeleteProgramInvocation-Request,
ENDIF
IF ( start )
 start                      [40] IMPLICIT Start-Request,
ENDIF
IF ( stop )
 stop                       [41] IMPLICIT Stop-Request,
ENDIF
IF ( resume )
 resume                     [42] IMPLICIT Resume-Request,
ENDIF
IF ( reset )
 reset                      [43] IMPLICIT Reset-Request,
ENDIF
IF ( kill )
 kill                       [44] IMPLICIT Kill-Request,
ENDIF
IF ( getProgramInvocationAttributes )
 getProgramInvocationAttributes [45] IMPLICIT
                        GetProgramInvocationAttributes-Request,
ENDIF
IF ( obtainFile )
 obtainFile                 [46] IMPLICIT ObtainFile-Request,
ENDIF
IF ( defineEventCondition )
 defineEventCondition       [47] IMPLICIT DefineEventCondition-Request,
ENDIF
IF ( deleteEventCondition )
 deleteEventCondition       [48] DeleteEventCondition-Request,
ENDIF
IF ( getEventConditionAttributes )
 getEventConditionAttributes  [49] GetEventConditionAttributes-Request,
ENDIF
IF ( reportEventConditionStatus )
 reportEventConditionStatus   [50] ReportEventConditionStatus-Request,
ENDIF
IF ( alterEventConditionMonitoring )
 alterEventConditionMonitoring   [51] IMPLICIT
                     AlterEventConditionMonitoring-Request,
ENDIF
IF ( triggerEvent )
 triggerEvent               [52] IMPLICIT TriggerEvent-Request,
ENDIF
IF ( defineEventAction )
```

```
 defineEventAction          [53] IMPLICIT DefineEventAction-Request,
ENDIF
IF ( deleteEventAction )
 deleteEventAction          [54] DeleteEventAction-Request,
ENDIF
IF ( getEventActionAttributes )
 getEventActionAttributes   [55] GetEventActionAttributes-Request,
ENDIF
IF ( reportEventActionStatus )
 reportEventActionStatus    [56] ReportEventActionStatus-Request,
ENDIF
IF ( defineEventEnrollment )
 defineEventEnrollment      [57] IMPLICIT DefineEventEnrollment-Request,
ENDIF
IF ( deleteEventEnrollment )
 deleteEventEnrollment      [58] DeleteEventEnrollment-Request,
ENDIF
IF ( alterEventEnrollment )
 alterEventEnrollment       [59] IMPLICIT AlterEventEnrollment-Request,
ENDIF
IF ( reportEventEnrollmentStatus )
 reportEventEnrollmentStatus  [60] ReportEventEnrollmentStatus-Request,
ENDIF
IF ( getEventEnrollmentAttributes )
 getEventEnrollmentAttributes [61] IMPLICIT
                     GetEventEnrollmentAttributes-Request,
ENDIF
IF ( acknowledgeEventNotification )
 acknowledgeEventNotification [62] IMPLICIT
                     AcknowledgeEventNotification-Request,
ENDIF
IF ( getAlarmSummary )
 getAlarmSummary            [63] IMPLICIT GetAlarmSummary-Request,
ENDIF
IF ( getAlarmEnrollmentSummary )
 getAlarmEnrollmentSummary [64] IMPLICIT GetAlarmEnrollmentSummary-Request,
ENDIF
IF ( readJournal )
 readJournal                [65] IMPLICIT ReadJournal-Request,
ENDIF
IF ( writeJournal )
 writeJournal               [66] IMPLICIT WriteJournal-Request,
ENDIF
IF ( initializeJournal )
 initializeJournal          [67] IMPLICIT InitializeJournal-Request,
ENDIF
IF ( reportJournalStatus )
 reportJournalStatus        [68] ReportJournalStatus-Request,
ENDIF
```

```
IF ( createJournal )
 createJournal              [69] IMPLICIT CreateJournal-Request,
ENDIF
IF ( deleteJournal )
 deleteJournal              [70] IMPLICIT DeleteJournal-Request,
ENDIF
IF ( getCapabilityList )
 getCapabilityList          [71] IMPLICIT GetCapabilityList-Request,
ENDIF
IF ( fileOpen )
 fileOpen                   [72] IMPLICIT FileOpen-Request,
ENDIF
IF ( fileRead )
 fileRead                   [73] IMPLICIT FileRead-Request,
ENDIF
IF ( fileClose )
 fileClose                  [74] IMPLICIT FileClose-Request,
ENDIF
IF ( fileRename )
 fileRename                 [75] IMPLICIT FileRename-Request,
ENDIF
IF ( fileDelete )
 fileDelete                 [76] IMPLICIT FileDelete-Request,
ENDIF
IF ( fileDirectory )
 fileDirectory              [77] IMPLICIT FileDirectory-Request,
ENDIF
IF ( csr cspi )
 additionalService          [78] AdditionalService-Request,
ENDIF
 -- choices [72] through [77] are reserved for use by services
 -- defined in annex C
 -- choice [79] is reserved
IF ( getDataExchangeAttributes )
 getDataExchangeAttributes [80] GetDataExchangeAttributes-Request,
                -- Shall not appear in minor version 1
ENDIF
IF ( exchangeData )
 exchangeData               [81] IMPLICIT ExchangeData-Request,
              -- Shall not appear in minor version 1
ENDIF
IF ( defineAccessControlList )
 defineAccessControlList   [82] IMPLICIT
                  DefineAccessControlList-Request,
              -- Shall not appear in minor version 1 or y
ENDIF
IF ( getAccessControlListAttributes )
 getAccessControlListAttributes  [83] IMPLICIT
                  GetAccessControlListAttributes-Request,
```

```
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( reportAccessControlledObjects )
 reportAccessControlledObjects   [84] IMPLICIT
                   ReportAccessControlledObjects-Request,
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( deleteAccessControlList )
 deleteAccessControlList   [85] IMPLICIT
                   DeleteAccessControlList-Request,
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( changeAccessControl )
 changeAccessControl       [86] IMPLICIT
                   ChangeAccessControl-Request
                -- Shall not appear in minor version 1 or y
ENDIF
 }
```

4.8 AdditionalService-Request

```
AdditionalService-Request ::= CHOICE {
IF ( csr )
IF ( vMDStop )
 vMDStop                   [0] IMPLICIT VMDStop-Request,
ENDIF
IF ( vMDReset )
 vMDReset                  [1] IMPLICIT VMDReset-Request,
ENDIF
IF ( select )
 select                    [2] IMPLICIT Select-Request,
ENDIF
IF ( alterProgramInvocationAttributes )
 alterPI                   [3] IMPLICIT AlterProgramInvocationAttributes-Request,
ENDIF
ENDIF
IF ( cspi )
IF ( initiateUnitControlLoad )
 initiateUCLoad            [4] IMPLICIT InitiateUnitControlLoad-Request,
ENDIF
IF ( unitControlLoadSegment )
 uCLoad                    [5] IMPLICIT UnitControlLoadSegment-Request,
ENDIF
IF ( unitControlUpload )
 uCUpload                  [6] IMPLICIT UnitControlUpload-Request,
ENDIF
IF ( startUnitControl )
 startUC                   [7] IMPLICIT StartUnitControl-Request,
ENDIF
```

```
IF ( stopUnitControl )
 stopUC                    [8] IMPLICIT StopUnitControl-Request,
ENDIF
IF ( createUnitControl )
 createUC                  [9] IMPLICIT CreateUnitControl-Request,
ENDIF
IF ( addToUnitControl )
 addToUC                   [10] IMPLICIT AddToUnitControl-Request,
ENDIF
IF ( removeFromUnitControl )
 removeFromUC              [11] IMPLICIT RemoveFromUnitControl-Request,
ENDIF
IF ( getUnitControlAttributes )
 getUCAttributes           [12] IMPLICIT GetUnitControlAttributes-Request,
ENDIF
IF ( loadUnitControlFromFile )
 loadUCFromFile            [13] IMPLICIT LoadUnitControlFromFile-Request,
ENDIF
IF ( storeUnitControlToFile )
 storeUCToFile             [14] IMPLICIT StoreUnitControlToFile-Request,
ENDIF
IF ( deleteUnitControl )
 deleteUC                  [15] IMPLICIT DeleteUnitControl-Request,
ENDIF
IF ( defineEventConditionList )
 defineECL                 [16] IMPLICIT DefineEventConditionList-Request,
ENDIF
IF ( deleteEventConditionList )
 deleteECL                 [17] IMPLICIT DeleteEventConditionList-Request,
ENDIF
IF ( addEventConditionListReference )
 addECLReference           [18] IMPLICIT AddEventConditionListReference-Request,
ENDIF
IF ( removeEventConditionListReference )
 removeECLReference        [19] IMPLICIT RemoveEventConditionListReference-Request,
ENDIF
IF ( getEventConditionListAttributes )
 getECLAttributes          [20] IMPLICIT GetEventConditionListAttributes-Request,
ENDIF
IF ( reportEventConditionListStatus )
 reportECLStatus           [21] IMPLICIT ReportEventConditionListStatus-Request,
ENDIF
IF ( alterEventConditionListMonitoring )
 alterECLMonitoring        [22] IMPLICIT AlterEventConditionListMonitoring-Request
ENDIF
ENDIF
 }
```

4.9 CS-Request-Detail

```
CS-Request-Detail ::= CHOICE {
otherRequests                NULL,
                  -- this choice shall be selected if the tag value of the
                  -- ConfirmedServiceRequest does not match any of the tags below
IF ( createProgramInvocation )
 createProgramInvocation   [38] IMPLICIT
                      CS-CreateProgramInvocation-Request,
ENDIF
IF ( start )
 start                     [40] IMPLICIT CS-Start-Request,
ENDIF
IF ( resume )
 resume                    [42] IMPLICIT CS-Resume-Request,
ENDIF
IF ( defineEventCondition )
 defineEventCondition      [47] IMPLICIT CS-DefineEventCondition-Request,
ENDIF
IF ( alterEventConditionMonitoring )
 alterEventConditionMonitoring   [51] IMPLICIT
                      CS-AlterEventConditionMonitoring-Request,
ENDIF
IF ( defineEventEnrollment )
 defineEventEnrollment     [57] IMPLICIT CS-DefineEventEnrollment-Request,
ENDIF
IF ( alterEventEnrollment )
 alterEventEnrollment      [59] IMPLICIT CS-AlterEventEnrollment-Request
ENDIF
 }
```

4.10 UnconfirmedService

```
UnconfirmedService ::= CHOICE {
IF (informationReport )
 informationReport         [0] IMPLICIT InformationReport,
ENDIF
IF ( unsolicitedStatus )
 unsolicitedStatus         [1] IMPLICIT UnsolicitedStatus,
ENDIF
IF ( eventNotification )
 eventNotification         [2] IMPLICIT EventNotification,
ENDIF
IF ( )
 additionalService         [3] AdditionalUnconfirmedService
ENDIF
 }
```

4.11 CS-Unconfirmed-Detail

```
CS-Unconfirmed-Detail ::= CHOICE {
IF ( cspi )
 eventNotification          [2] IMPLICIT CS-EventNotification,
ENDIF
 otherRequests              NULL
      -- this choice shall be selected if the tag value of the
      -- UnconfirmedService does not match any of the tags above
 }
```

4.12 ConfirmedServiceResponse

```
ConfirmedServiceResponse ::= CHOICE {
IF ( status )
 status                     [0] IMPLICIT Status-Response,
ENDIF
IF ( getNameList )
 getNameList                [1] IMPLICIT GetNameList-Response,
ENDIF
IF ( identify )
 identify                   [2] IMPLICIT Identify-Response,
ENDIF
IF ( rename )
 rename                     [3] IMPLICIT Rename-Response,
ENDIF
IF ( read )
 read                       [4] IMPLICIT Read-Response,
ENDIF
IF ( write )
 write                      [5] IMPLICIT Write-Response,
ENDIF
IF ( vnam vadr )
IF ( getVariableAccessAttributes )
 getVariableAccessAttributes  [6] IMPLICIT
                    GetVariableAccessAttributes-Response,
ENDIF
ENDIF
IF ( vnam )
IF ( vadr )
IF ( defineNamedVariable )
 defineNamedVariable        [7] IMPLICIT DefineNamedVariable-Response,
ENDIF
ENDIF
ENDIF
IF ( vsca )
IF ( vnam vadr )
IF ( defineScatteredAccess )
 defineScatteredAccess      [8] IMPLICIT DefineScatteredAccess-Response,
```

```
ENDIF
IF ( getScatteredAccessAttributes )
 getScatteredAccessAttributes [9] IMPLICIT
                     GetScatteredAccessAttributes-Response,
ENDIF
ENDIF
ENDIF
IF ( vsca vnam )
IF ( deleteVariableAccess )
 deleteVariableAccess       [10] IMPLICIT DeleteVariableAccess-Response,
ENDIF
ENDIF
IF ( vlis )
IF ( vnam )
IF ( defineNamedVariableList )
 defineNamedVariableList    [11] IMPLICIT DefineNamedVariableList-Response,
ENDIF
IF ( getNamedVariableListAttributes )
 getNamedVariableListAttributes [12] IMPLICIT
                     GetNamedVariableListAttributes-Response,
ENDIF
IF ( deleteNamedVariableList )
 deleteNamedVariableList    [13] IMPLICIT DeleteNamedVariableList-Response,
ENDIF
ENDIF
ENDIF
IF ( vnam )
IF ( defineNamedType )
 defineNamedType            [14] IMPLICIT DefineNamedType-Response,
ENDIF
IF ( getNamedTypeAttributes )
 getNamedTypeAttributes     [15] IMPLICIT GetNamedTypeAttributes-Response,
ENDIF
IF ( deleteNamedType )
 deleteNamedType            [16] IMPLICIT DeleteNamedType-Response,
ENDIF
ENDIF
IF ( input )
 input                      [17] IMPLICIT Input-Response,
ENDIF
IF ( output )
 output                     [18] IMPLICIT Output-Response,
ENDIF
IF ( takeControl )
 takeControl                [19] TakeControl-Response,
ENDIF
IF ( relinquishControl )
 relinquishControl          [20] IMPLICIT RelinquishControl-Response,
ENDIF
```

```
IF ( defineSemaphore )
 defineSemaphore            [21] IMPLICIT DefineSemaphore-Response,
ENDIF
IF ( deleteSemaphore )
 deleteSemaphore            [22] IMPLICIT DeleteSemaphore-Response,
ENDIF
IF ( reportSemaphoreStatus )
 reportSemaphoreStatus      [23] IMPLICIT ReportSemaphoreStatus-Response,
ENDIF
IF ( reportPoolSemaphoreStatus )
 reportPoolSemaphoreStatus [24] IMPLICIT ReportPoolSemaphoreStatus-Response,
ENDIF
IF ( reportSemaphoreEntryStatus )
 reportSemaphoreEntryStatus   [25] IMPLICIT ReportSemaphoreEntryStatus-Response,
ENDIF
IF ( initiateDownloadSequence )
 initiateDownloadSequence  [26] IMPLICIT InitiateDownloadSequence-Response,
 downloadSegment           [27] IMPLICIT DownloadSegment-Response,
 terminateDownloadSequence [28] IMPLICIT
                    TerminateDownloadSequence-Response,
ENDIF
IF ( initiateUploadSequence )
 initiateUploadSequence     [29] IMPLICIT InitiateUploadSequence-Response,
 uploadSegment              [30] IMPLICIT UploadSegment-Response,
 terminateUploadSequence    [31] IMPLICIT TerminateUploadSequence-Response,
ENDIF
IF ( requestDomainDownload )
 requestDomainDownload      [32] IMPLICIT RequestDomainDownload-Response,
ENDIF
IF ( requestDomainUpload )
 requestDomainUpload        [33] IMPLICIT RequestDomainUpload-Response,
ENDIF
IF ( loadDomainContent )
 loadDomainContent          [34] IMPLICIT LoadDomainContent-Response,
ENDIF
IF ( storeDomainContent )
 storeDomainContent         [35] IMPLICIT StoreDomainContent-Response,
ENDIF
IF ( deleteDomain )
 deleteDomain               [36] IMPLICIT DeleteDomain-Response,
ENDIF
IF ( getDomainAttributes )
 getDomainAttributes        [37] IMPLICIT GetDomainAttributes-Response,
ENDIF
IF ( createProgramInvocation )
 createProgramInvocation    [38] IMPLICIT CreateProgramInvocation-Response,
ENDIF
IF ( deleteProgramInvocation )
 deleteProgramInvocation    [39] IMPLICIT DeleteProgramInvocation-Response,
```

```
ENDIF
IF ( start )
 start                      [40] IMPLICIT Start-Response,
ENDIF
IF ( stop )
 stop                       [41] IMPLICIT Stop-Response,
ENDIF
IF ( resume )
 resume                     [42] IMPLICIT Resume-Response,
ENDIF
IF ( reset )
 reset                      [43] IMPLICIT Reset-Response,
ENDIF
IF ( kill )
 kill                       [44] IMPLICIT Kill-Response,
ENDIF
IF ( getProgramInvocationAttributes )
 getProgramInvocationAttributes [45] IMPLICIT
                    GetProgramInvocationAttributes-Response,
ENDIF
IF ( obtainFile )
 obtainFile                 [46] IMPLICIT ObtainFile-Response,
ENDIF
IF ( defineEventCondition )
 defineEventCondition       [47] IMPLICIT DefineEventCondition-Response,
ENDIF
IF ( deleteEventCondition )
 deleteEventCondition       [48] IMPLICIT DeleteEventCondition-Response,
ENDIF
IF ( getEventConditionAttributes )
 getEventConditionAttributes [49] IMPLICIT
                    GetEventConditionAttributes-Response,
ENDIF
IF ( reportEventConditionStatus )
 reportEventConditionStatus   [50] IMPLICIT
                    ReportEventConditionStatus-Response,
ENDIF
IF ( alterEventConditionMonitoring )
 alterEventConditionMonitoring   [51] IMPLICIT
                    AlterEventConditionMonitoring-Response,
ENDIF
IF ( triggerEvent )
 triggerEvent               [52] IMPLICIT TriggerEvent-Response,
ENDIF
IF ( defineEventAction )
 defineEventAction          [53] IMPLICIT DefineEventAction-Response,
ENDIF
IF ( deleteEventAction )
 deleteEventAction          [54] IMPLICIT DeleteEventAction-Response,
```

```
ENDIF
IF ( getEventActionAttributes )
 getEventActionAttributes  [55] IMPLICIT GetEventActionAttributes-Response,
ENDIF
IF ( reportEventActionStatus )
 reportEventActionStatus   [56] IMPLICIT ReportEventActionStatus-Response,
ENDIF
IF ( defineEventEnrollment )
 defineEventEnrollment     [57] IMPLICIT DefineEventEnrollment-Response,
ENDIF
IF ( deleteEventEnrollment )
 deleteEventEnrollment     [58] IMPLICIT DeleteEventEnrollment-Response,
ENDIF
IF ( alterEventEnrollment )
 alterEventEnrollment      [59] IMPLICIT AlterEventEnrollment-Response,
ENDIF
IF ( reportEventEnrollmentStatus )
 reportEventEnrollmentStatus  [60] IMPLICIT
                      ReportEventEnrollmentStatus-Response,
ENDIF
IF ( getEventEnrollmentAttributes )
 getEventEnrollmentAttributes [61] IMPLICIT
                      GetEventEnrollmentAttributes-Response,
ENDIF
IF ( acknowledgeEventNotification )
 acknowledgeEventNotification [62] IMPLICIT
                      AcknowledgeEventNotification-Response,
ENDIF
IF ( getAlarmSummary )
 getAlarmSummary           [63] IMPLICIT GetAlarmSummary-Response,
ENDIF
IF ( getAlarmEnrollmentSummary )
 getAlarmEnrollmentSummary [64] IMPLICIT
                      GetAlarmEnrollmentSummary-Response,
ENDIF
IF ( readJournal )
 readJournal               [65] IMPLICIT ReadJournal-Response,
ENDIF
IF ( writeJournal )
 writeJournal              [66] IMPLICIT WriteJournal-Response,
ENDIF
IF ( initializeJournal )
 initializeJournal         [67] IMPLICIT InitializeJournal-Response,
ENDIF
IF ( reportJournalStatus )
 reportJournalStatus       [68] IMPLICIT ReportJournalStatus-Response,
ENDIF
IF ( createJournal )
 createJournal             [69] IMPLICIT CreateJournal-Response,
```

```
ENDIF
IF ( deleteJournal )
 deleteJournal              [70] IMPLICIT DeleteJournal-Response,
ENDIF
IF ( getCapabilityList )
 getCapabilityList          [71] IMPLICIT GetCapabilityList-Response,
ENDIF
IF ( fileOpen )
 fileOpen                   [72] IMPLICIT FileOpen-Response,
ENDIF
IF ( fileRead )
 fileRead                   [73] IMPLICIT FileRead-Response,
ENDIF
IF ( fileClose )
 fileClose                  [74] IMPLICIT FileClose-Response,
ENDIF
IF ( fileRename )
 fileRename                 [75] IMPLICIT FileRename-Response,
ENDIF
IF ( fileDelete )
 fileDelete                 [76] IMPLICIT FileDelete-Response,
ENDIF
IF ( fileDirectory )
 fileDirectory              [77] IMPLICIT FileDirectory-Response,
ENDIF
IF ( csr cspi )
 additionalService          [78] AdditionalService-Response,
ENDIF
 -- choices [72] through [77] are reserved for use by services
 -- defined in annex C
 -- choice [79] is reserved
IF ( getDataExchangeAttributes )
 getDataExchangeAttributes [80] GetDataExchangeAttributes-Response,
                -- Shall not appear in minor version 1
ENDIF
IF ( exchangeData )
 exchangeData               [81] IMPLICIT ExchangeData-Response,
                -- Shall not appear in minor version 1
ENDIF
IF ( defineAccessControlList )
 defineAccessControlList    [82] IMPLICIT
                   DefineAccessControlList-Response,
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( getAccessControlListAttributes )
 getAccessControlListAttributes  [83] IMPLICIT
                   GetAccessControlListAttributes-Response,
                -- Shall not appear in minor version 1 or y
ENDIF
```

```
IF ( reportAccessControlledObjects )
 reportAccessControlledObjects   [84] IMPLICIT
                  ReportAccessControlledObjects-Response,
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( deleteAccessControlList )
 deleteAccessControlList   [85] IMPLICIT
                  DeleteAccessControlList-Response,
                -- Shall not appear in minor version 1 or y
ENDIF
IF ( changeAccessControl )
 changeAccessControl       [86] IMPLICIT ChangeAccessControl-Response
                -- Shall not appear in minor version 1 or y
ENDIF
 }
```

4.13 AdditionalService-Response

```
AdditionalService-Response ::= CHOICE {
IF ( csr )
IF ( vMDStop )
 vMDStop                    [0] IMPLICIT VMDStop-Response,
ENDIF
IF ( vMDReset )
 vMDReset                   [1] IMPLICIT VMDReset-Response,
ENDIF
IF ( select )
 select                     [2] IMPLICIT Select-Response,
ENDIF
IF ( alterProgramInvocationAttributes )
 alterPI                    [3] IMPLICIT
              AlterProgramInvocationAttributes-Response,
ENDIF
ENDIF
IF ( cspi )
IF ( initiateUnitControlLoad )
 initiateUCLoad             [4] IMPLICIT InitiateUnitControlLoad-Response,
ENDIF
IF ( unitControlLoadSegment )
 uCLoad                     [5] IMPLICIT UnitControlLoadSegment-Response,
ENDIF
IF ( unitControlUpload )
 uCUpload                   [6] IMPLICIT UnitControlUpload-Response,
ENDIF
IF ( startUnitControl )
 startUC                    [7] IMPLICIT StartUnitControl-Response,
ENDIF
IF ( stopUnitControl )
 stopUC                     [8] IMPLICIT StopUnitControl-Response,
```

```
ENDIF
IF ( createUnitControl )
 createUC                  [9] IMPLICIT CreateUnitControl-Response,
ENDIF
IF ( addToUnitControl )
 addToUC                   [10] IMPLICIT AddToUnitControl-Response,
ENDIF
IF ( removeFromUnitControl )
 removeFromUC              [11] IMPLICIT RemoveFromUnitControl-Response,
ENDIF
IF ( getUnitControlAttributes )
 getUCAttributes           [12] IMPLICIT GetUnitControlAttributes-Response,
ENDIF
IF ( loadUnitControlFromFile )
 loadUCFromFile            [13] IMPLICIT LoadUnitControlFromFile-Response,
ENDIF
IF ( storeUnitControlToFile )
 storeUCToFile             [14] IMPLICIT StoreUnitControlToFile-Response,
ENDIF
IF ( deleteUnitControl )
 deleteUC                  [15] IMPLICIT DeleteUnitControl-Response,
ENDIF
IF ( defineEventConditionList )
 defineECL                 [16] IMPLICIT DefineEventConditionList-Response,
ENDIF
IF ( deleteEventConditionList )
 deleteECL                 [17] IMPLICIT DeleteEventConditionList-Response,
ENDIF
IF ( addEventConditionListReference )
 addECLReference           [18] IMPLICIT AddEventConditionListReference-Response,
ENDIF
IF ( removeEventConditionListReference )
 removeECLReference        [19] IMPLICIT
              RemoveEventConditionListReference-Response,
ENDIF
IF ( getEventConditionListAttributes )
 getECLAttributes      [20] IMPLICIT GetEventConditionListAttributes-Response,
ENDIF
IF ( reportEventConditionListStatus )
 reportECLStatus           [21] IMPLICIT ReportEventConditionListStatus-Response,
ENDIF
IF ( alterEventConditionListMonitoring )
 alterECLMonitoring    [22] IMPLICIT AlterEventConditionListMonitoring-Response
ENDIF
ENDIF
 }
```

4.14 CS-Response-Detail

```
CS-Response-Detail ::= CHOICE {
otherRequests                NULL,
                  -- this choice shall be selected if the tag value of the
                  -- ConfirmedServiceResponse does not match any of the tags
below
IF ( status )
 status                      [0] IMPLICIT CS-Status-Response,
ENDIF
IF ( getProgramInvocationAttributes )
 getProgramInvocationAttributes [45] IMPLICIT
                  CS-GetProgramInvocationAttributes-Response,
ENDIF
IF ( defineEventCondition )
 defineEventCondition        [47] IMPLICIT CS-DefineEventCondition-Response,
ENDIF
IF ( getEventConditionAttributes )
 getEventConditionAttributes  [49] IMPLICIT
                     CS-GetEventConditionAttributes-Response,
ENDIF
IF ( defineEventEnrollment )
 defineEventEnrollment       [57] IMPLICIT CS-DefineEventEnrollment-Response
ENDIF
 }
```

4.15 ServiceError

```
ServiceError ::= SEQUENCE {
 errorClass          [0] CHOICE {
 vmd-state                 [0] IMPLICIT INTEGER { -- VMD-STATE,
   other                           (0), -- OTHER
   vmd-state-conflict              (1), -- VMD-STATE-CONFLICT
   vmd-operational-problem         (2), -- VMD-OPERATIONAL-PROBLEM
   domain-transfer-problem         (3), -- DOMAIN-TRANSFER-PROBLEM
   state-machine-id-invalid        (4) -- STATE-MACHINE-ID-INVALID
 },
 application-reference  [1] IMPLICIT INTEGER { -- APPLICATION REFERENCE
   other                           (0), -- OTHER
   application-unreachable         (1), -- APPLICATION-UNREACHABLE
   connection-lost                 (2), -- CONNECTION-LOST
   application-reference-invalid   (3), -- APPLICATION-REFERENCE-INVALID
   context-unsupported             (4) -- CONTEXT-UNSUPPORTED
 },
 definition              [2] IMPLICIT INTEGER { -- DEFINITION
   other                           (0), -- OTHER
   object-undefined                (1), -- OBJECT-UNDEFINED
   invalid-address                 (2), -- INVALID-ADDRESS
   type-unsupported                (3), -- TYPE-UNSUPPORTED
```

```
    type-inconsistent                (4), -- TYPE-INCONSISTENT
    object-exists                    (5), -- OBJECT-EXISTS
    object-attribute-inconsistent    (6)   -- OBJECT-ATTRIBUTE-INCONSISTENT
  },
  resource                [3] IMPLICIT INTEGER { -- RESOURCE
    other                            (0), -- OTHER
    memory-unavailable               (1), -- MEMORY-UNAVAILABLE
    processor-resource-unavailable   (2), -- PROCESSOR-RESOURCE-UNAVAILABLE
    mass-storage-unavailable         (3), -- MASS-STORAGE-UNAVAILABLE
    capability-unavailable           (4), -- CAPABILITY-UNAVAILABLE
    capability-unknown               (5)   -- CAPABILITY-UNKNOWN
  },
  service                 [4] IMPLICIT INTEGER { -- SERVICE
    other                            (0), -- OTHER
    primitives-out-of-sequence       (1), -- PRIMITIVES-OUT-OF-SEQUENCE
    object-state-conflict            (2), -- OBJECT-STATE-CONFLICT
                  -- Value 3 reserved for further definition
    continuation-invalid             (4), -- CONTINUATION-INVALID
    object-constraint-conflict       (5)   -- OBJECT-CONSTRAINT-CONFLICT
  },
  service-preempt         [5] IMPLICIT INTEGER { -- SERVICE-PREEMPT
    other                            (0), -- OTHER
    timeout                          (1), -- TIMEOUT
    deadlock                         (2), -- DEADLOCK
    cancel                           (3)   -- CANCEL
  },
  time-resolution         [6] IMPLICIT INTEGER { -- TIME-RESOLUTION
    other                            (0), -- OTHER
    unsupportable-time-resolution    (1)   -- UNSUPPORTABLE-TIME-RESOLUTION
  },
  access                  [7] IMPLICIT INTEGER { -- ACCESS
    other                            (0), -- OTHER
    object-access-unsupported        (1), -- OBJECT-ACCESS-UNSUPPORTED
    object-non-existent              (2), -- OBJECT-NON-EXISTENT
    object-access-denied             (3), -- OBJECT-ACCESS-DENIED
    object-invalidated               (4)   -- OBJECT-INVALIDATED
  },
  initiate                [8] IMPLICIT INTEGER { -- INITIATE
    other                            (0), -- OTHER
                -- Values 1 and 2 are reserved for further definition
    max-services-outstanding-calling-insufficient (3),
        -- MAX-SERVICES-OUTSTANDING-CALLING-INSUFFICIENT
    max-services-outstanding-called-insufficient (4),
          -- MAX-SERVICES-OUTSTANDING-CALLED-INSUFFICIENT
    service-CBB-insufficient         (5), -- SERVICE-CBB-INSUFFICIENT
    parameter-CBB-insufficient       (6), -- PARAMETER-CBB-INSUFFICIENT
    nesting-level-insufficient       (7)   -- NESTING-LEVEL-INSUFFICIENT
  },
  conclude                [9] IMPLICIT INTEGER { -- CONCLUDE
```

```
   other                                (0), -- OTHER
   further-communication-required       (1) -- FURTHER-COMMUNICATION-REQUIRED
 },
IF ( cancel )
 cancel                    [10] IMPLICIT INTEGER { -- CANCEL
   other                                (0),  -- OTHER
   invoke-id-unknown                    (1),  -- INVOKE-ID-UNKNOWN
   cancel-not-possible                  (2)   -- CANCEL-NOT-POSSIBLE
 },
ENDIF
IF ( fileOpen fileClose fileRead fileRename fileDelete fileDirectory obtainFile
)
 file                      [11] IMPLICIT INTEGER { -- FILE
   other                                (0),  -- OTHER
   filename-ambiguous                   (1), -- FILENAME-AMBIGUOUS
   file-busy                            (2),  -- FILE-BUSY
   filename-syntax-error                (3), -- FILENAME-SYNTAX-ERROR
   content-type-invalid                 (4), -- CONTENT-TYPE-INVALID
   position-invalid                     (5), -- POSITION-INVALID
   file-access-denied                   (6), -- FILE-ACCESS-DENIED
   file-non-existent                    (7), -- FILE-NON-EXISTENT
   duplicate-filename                   (8), -- DUPLICATE-FILENAME
   insufficient-space-in-filestore      (9) -- INSUFFICIENT-SPACE-IN-FILESTORE
 },
ENDIF
 others                    [12] IMPLICIT INTEGER, -- OTHERS
IF ( cspi )
 cs-error                  [13] CS-Service-Error
ENDIF
 } ,
 additionalCode    [1] IMPLICIT INTEGER OPTIONAL,
 additionalDescription  [2] IMPLICIT VisibleString OPTIONAL,
IF    (    obtainFile    start    stop    resume    reset    deleteVariableAccess
deleteNamedVariableList deleteNamedType
defineEventEnrollment fileRename )
 serviceSpecificInformation    [3] CHOICE {
IF ( obtainFile )
 obtainFile                    [0] IMPLICIT ObtainFile-Error,
ENDIF
IF ( start )
 start                         [1] IMPLICIT Start-Error,
ENDIF
IF ( stop )
 stop                          [2] IMPLICIT Stop-Error,
ENDIF
IF ( resume )
 resume                        [3] IMPLICIT Resume-Error,
ENDIF
IF ( reset )
```

```
 reset                          [4] IMPLICIT Reset-Error,
ENDIF
IF ( deleteVariableAccess )
 deleteVariableAccess           [5] IMPLICIT DeleteVariableAccess-Error,
ENDIF
IF ( deleteNamedVariableList )
 deleteNamedVariableList        [6] IMPLICIT
                      DeleteNamedVariableList-Error,
ENDIF
IF ( deleteNamedType )
 deleteNamedType                [7] IMPLICIT DeleteNamedType-Error,
ENDIF
IF ( defineEventEnrollment )
 defineEventEnrollment-Error    [8] DefineEventEnrollment-Error,
ENDIF
IF ( fileRename )
 fileRename                     [9] IMPLICIT FileRename-Error,
      -- Reserved for use by annex C Rename service
ENDIF
IF ( csr cspi )
 additionalService              [10] AdditionalService-Error,
ENDIF
   changeAccessControl          [11] IMPLICIT ChangeAccessControlError
   } OPTIONAL
ENDIF
 }
```

4.16 AdditionalService-Error

```
AdditionalService-Error ::= CHOICE {
IF ( cspi )
IF ( defineEventConditionList )
 defineEcl                  [0] DefineEventConditionList-Error,
ENDIF
IF ( addEventConditionListReference )
 addECLReference            [1] AddEventConditionListReference-Error,
ENDIF
IF ( removeEventConditionListReference )
 removeECLReference         [2] RemoveEventConditionListReference-Error,
ENDIF
IF ( initiateUnitControlLoad )
 initiateUC                 [3] InitiateUnitControl-Error,
ENDIF
IF ( startUnitControl )
 startUC                    [4] IMPLICIT StartUnitControl-Error,
ENDIF
IF ( stopUnitControl )
 stopUC                     [5] IMPLICIT StopUnitControl-Error,
ENDIF
```

```
IF ( deleteUnitControl )
 deleteUC                [6] IMPLICIT DeleteUnitControl-Error,
ENDIF
IF ( loadUnitControlFromFile )
 loadUCFromFile          [7] IMPLICIT LoadUnitControlFromFile-Error
ENDIF
ENDIF
 }
```

4.17 TimeOfDay

```
TimeOfDay ::= OCTET STRING (SIZE(4|6))
```

4.18 Identifier

```
Identifier ::= VisibleString (FROM
 ("A"|"a"|"B"|"b"|"C"|"c"|"D"|"d"|"E"|"e"|"F"|"f"|
  "G"|"g"|"H"|"h"|"I"|"i"|"J"|"j"|"K"|"k"|"L"|"l"|
  "M"|"m"|"N"|"n"|"O"|"o"|"P"|"p"|"Q"|"q"|"R"|"r"|
  "S"|"s"|"T"|"t"|"U"|"u"|"V"|"v"|"W"|"w"|"X"|"x"|
  "Y"|"y"|"Z"|"z"|"$"|"_"|"0"|"1"|"2"|"3"|"4"|"5"|
  "6"|"7"|"8"|"9")) ( SIZE(1..32))
   -- An Identifier shall not begin with a digit.
```

4.19 Integer's

```
Integer8 ::= INTEGER(-128..127)          -- range -128 <= i <= 127

Integer16 ::= INTEGER(-32768..32767)     -- range -32,768 <= i <= 32,767

Integer32 ::= INTEGER(-2147483648..2147483647)
                                 -- range -2**31 <= i <= 2**31 - 1

Unsigned8 ::= INTEGER(0..127)            -- range 0 <= i <= 127

Unsigned16 ::= INTEGER(0..32767)         -- range 0 <= i <= 32767

Unsigned32 ::= INTEGER(0..2147483647)    -- range 0 <= i <= 2**31 - 1
```

4.20 ObjectName

```
ObjectName ::= CHOICE {
 vmd-specific               [0] IMPLICIT Identifier,
 domain-specific            [1] IMPLICIT SEQUENCE {
 domainID                         Identifier,
 itemID                           Identifier },
 aa-specific                [2] IMPLICIT Identifier }
```

4.21 ApplicationReference

```
ApplicationReference ::= SEQUENCE {
 ap-title                  [0] ISO-8650-ACSE-1.AP-title OPTIONAL,
 ap-invocation-id          [1] ISO-8650-ACSE-1.AP-invocation-id OPTIONAL,
 ae-qualifier              [2] ISO-8650-ACSE-1.AE-qualifier OPTIONAL,
 ae-invocation-id          [3] ISO-8650-ACSE-1.AE-invocation-id OPTIONAL }
```

4.22 FileName

```
FileName ::= SEQUENCE OF GraphicString
```

4.23 Priority

```
Priority ::= Unsigned8

normalPriority Priority ::= 64
```

4.24 Severity

```
Severity ::= Unsigned8

normalSeverity Severity ::= 64
```

4.25 Initiate-PDU's

```
Initiate-RequestPDU ::= SEQUENCE {
 localDetailCalling                    [0] IMPLICIT Integer32 OPTIONAL,
 proposedMaxServOutstandingCalling     [1] IMPLICIT Integer16,
 proposedMaxServOutstandingCalled      [2] IMPLICIT Integer16,
 proposedDataStructureNestingLevel     [3] IMPLICIT Integer8 OPTIONAL,
 initRequestDetail                     [4] IMPLICIT InitRequestDetail }

Initiate-ResponsePDU ::= SEQUENCE {
 localDetailCalled                     [0] IMPLICIT Integer32 OPTIONAL,
 negotiatedMaxServOutstandingCalling   [1] IMPLICIT Integer16,
 negotiatedMaxServOutstandingCalled    [2] IMPLICIT Integer16,
 negotiatedDataStructureNestingLevel   [3] IMPLICIT Integer8 OPTIONAL,
 initResponseDetail                    [4] IMPLICIT InitResponseDetail }

Initiate-ErrorPDU ::= ServiceError
```

4.26 Initiate-Detail's

```
InitRequestDetail ::= SEQUENCE {
 proposedVersionNumber              [0] IMPLICIT Integer16,
 proposedParameterCBB               [1] IMPLICIT ParameterSupportOptions,
 servicesSupportedCalling           [2] IMPLICIT ServiceSupportOptions ,
IF (csr cspi)
 additionalSupportedCalling         [3] IMPLICIT AdditionalSupportOptions,
ENDIF
IF (cspi)
 additionalCbbSupportedCalling      [4] IMPLICIT AdditionalCbbOptions,
 privilegeClassIdentityCalling      [5] IMPLICIT VisibleString
ENDIF
 }

InitResponseDetail ::= SEQUENCE {
 negotiatedVersionNumber            [0] IMPLICIT Integer16,
 negotiatedParameterCBB             [1] IMPLICIT ParameterSupportOptions,
 servicesSupportedCalled            [2] IMPLICIT ServiceSupportOptions,
IF (csr cspi)
 additionalSupportedCalled          [3] IMPLICIT AdditionalSupportOptions,
ENDIF
IF (cspi )
 additionalCbbSupportedCalled       [4] IMPLICIT AdditionalCbbOptions,
 privilegeClassIdentityCalled       [5] IMPLICIT VisibleString
ENDIF
 }
```

4.27 ServiceSupportOptions

```
ServiceSupportOptions ::= BIT STRING {
 status                          (0),
 getNameList                     (1),
 identify                        (2),
 rename                          (3),
 read                            (4),
 write                           (5),
 getVariableAccessAttributes     (6),
 defineNamedVariable             (7),
 defineScatteredAccess           (8),
 getScatteredAccessAttributes    (9),
 deleteVariableAccess            (10),
 defineNamedVariableList         (11),
 getNamedVariableListAttributes  (12),
 deleteNamedVariableList         (13),
 defineNamedType                 (14),
 getNamedTypeAttributes          (15),
 deleteNamedType                 (16),
 input                           (17),
```

```
output                          (18),
takeControl                     (19),
relinquishControl               (20),
defineSemaphore                 (21),
deleteSemaphore                 (22),
reportSemaphoreStatus           (23),
reportPoolSemaphoreStatus       (24),
reportSemaphoreEntryStatus      (25),
initiateDownloadSequence        (26),
downloadSegment                 (27),
terminateDownloadSequence       (28),
initiateUploadSequence          (29),
uploadSegment                   (30),
terminateUploadSequence         (31),
requestDomainDownload           (32),
requestDomainUpload             (33),
loadDomainContent               (34),
storeDomainContent              (35),
deleteDomain                    (36),
getDomainAttributes             (37),
createProgramInvocation         (38),
deleteProgramInvocation         (39),
start                           (40),
stop                            (41),
resume                          (42),
reset                           (43),
kill                            (44),
getProgramInvocationAttributes  (45),
obtainFile                      (46),
defineEventCondition            (47),
deleteEventCondition            (48),
getEventConditionAttributes     (49),
reportEventConditionStatus      (50),
alterEventConditionMonitoring   (51),
triggerEvent                    (52),
defineEventAction               (53),
deleteEventAction               (54),
getEventActionAttributes        (55),
reportEventActionStatus         (56),
defineEventEnrollment           (57),
deleteEventEnrollment           (58),
alterEventEnrollment            (59),
reportEventEnrollmentStatus     (60),
getEventEnrollmentAttributes    (61),
acknowledgeEventNotification    (62),
getAlarmSummary                 (63),
getAlarmEnrollmentSummary       (64),
readJournal                     (65),
writeJournal                    (66),
```

```
 initializeJournal                (67),
 reportJournalStatus              (68),
 createJournal                    (69),
 deleteJournal                    (70),
 getCapabilityList                (71),
 fileOpen              (72), -- reserved for use of services defined in annex C
 fileRead              (73), -- reserved for use of services defined in annex C
 fileClose             (74), -- reserved for use of services defined in annex C
 fileRename            (75), -- reserved for use of services defined in annex C
 fileDelete            (76), -- reserved for use of services defined in annex C
 fileDirectory         (77), -- reserved for use of services defined in annex C
 unsolicitedStatus                (78),
 informationReport                (79),
 eventNotification                (80),
 attachToEventCondition           (81),
 attachToSemaphore                (82),
 conclude                         (83),
 cancel                           (84),
 getDataExchangeAttributes        (85),
      -- Shall not appear in minor version 1
 exchangeData                     (86),
      -- Shall not appear in minor version 1
 defineAccessControlList          (87),
      -- Shall not appear in minor version 1 or y
 getAccessControlListAttributes   (88),
      -- Shall not appear in minor version 1 or y
 reportAccessControlledObjects    (89),
      -- Shall not appear in minor version 1 or y
 deleteAccessControlList          (90),
      -- Shall not appear in minor version 1 or y
 changeAccessControl              (91)
      -- Shall not appear in minor version 1 or y
}

AdditionalSupportOptions ::= BITSTRING {
IF (csr)
 vMDStop                                (0),
 vMDReset                               (1),
 select                                 (2),
 alterProgramInvocationAttributes       (3),
ENDIF
IF (cspi)
 initiateUnitControlLoad                (4),
 unitControlLoadSegment                 (5),
 unitControlUpload                      (6),
 startUnitControl                       (7),
 stopUnitControl                        (8),
 createUnitControl                      (9),
 addToUnitControl                       (10),
```

```
 removeFromUnitControl              (11),
 getUnitControlAttributes           (12),
 loadUnitControlFromFile            (13),
 storeUnitControlToFile             (14),
 deleteUnitControl                  (15),
 defineEventConditionList           (16),
 deleteEventConditionList           (17),
 addEventConditionListReference     (18),
 removeEventConditionListReference  (19),
 getEventConditionListAttributes    (20),
 reportEventConditionListStatus     (21),
 alterEventConditionListMonitoring  (22)
ENDIF
 }
```

4.28 ParameterSupportOptions

```
ParameterSupportOptions ::= BIT STRING {
 str1                           (0),
 str2                           (1),
 vnam                           (2),
 valt                           (3),
 vadr                           (4),
 vsca                           (5),
 tpy                            (6),
 vlis                           (7),
 real                           (8),
      -- bit 9 is reserved for future definition
 cei                            (10),
 acu                            (11),
 aco                            (12)
      -- shall not be set unless acu is also set.
 }

AdditionalCbbOptions ::= BITSTRING {
 des                            (0),
 dei                            (1),
 recl                           (2) }
```

4.29 Conclude-PDU's

```
Conclude-RequestPDU ::= NULL

Conclude-ResponsePDU ::= NULL
```

```
Conclude-ErrorPDU ::= ServiceError
```

4.30 Cancel-PDU's

```
Cancel-RequestPDU ::= Unsigned32 -- originalInvokeID

Cancel-ResponsePDU ::= Unsigned32 -- originalInvokeID

Cancel-ErrorPDU ::= SEQUENCE {
 originalInvokeID          [0] IMPLICIT Unsigned32,
 serviceError              [1] IMPLICIT ServiceError }
```

4.31 RejectPDU

```
RejectPDU ::= SEQUENCE {
 originalInvokeID          [0] IMPLICIT Unsigned32 OPTIONAL,
 rejectReason              CHOICE {
 confirmed-requestPDU         [1] IMPLICIT INTEGER { -- CONFIRMED-REQUESTPDU
   other                         (0), -- OTHER
   unrecognized-service          (1), -- UNRECOGNIZED-SERVICE
   unrecognized-modifier         (2), -- UNRECOGNIZED-MODIFIER
   invalid-invokeID              (3), -- INVALID-INVOKEID
   invalid-argument              (4), -- INVALID-ARGUMENT
   invalid-modifier              (5), -- INVALID-MODIFIER
   max-serv-outstanding-exceeded (6), -- MAX-SERV-OUTSTANDING-EXCEEDED
                      -- Value 7 reserved for further definition
   max-recursion-exceeded        (8), -- MAX-RECURSION-EXCEEDED
   value-out-of-range            (9) -- VALUE-OUT-OF-RANGE
 },
 confirmed-responsePDU       [2] IMPLICIT INTEGER { --CONFIRMED-RESPONSEPDU
   other                         (0), -- OTHER
   unrecognized-service          (1), -- UNRECOGNIZED-SERVICE
   invalid-invokeID              (2), -- INVALID-INVOKEID
   invalid-result                (3), -- INVALID-RESULT
                  -- Value 4 reserved for further definition
   max-recursion-exceeded        (5), -- MAX-RECURSION-EXCEEDED
   value-out-of-range            (6) -- VALUE-OUT-OF-RANGE
 },
 confirmed-errorPDU          [3] IMPLICIT INTEGER { -- CONFIRMED-ERRORPDU
   other                         (0), -- OTHER
   unrecognized-service          (1), -- UNRECOGNIZED-SERVICE
   invalid-invokeID              (2), -- INVALID-INVOKEID
   invalid-serviceError          (3), -- INVALID-SERVICEERROR
   value-out-of-range            (4) -- VALUE-OUT-OF-RANGE
 },
 unconfirmedPDU              [4] IMPLICIT INTEGER { -- UNCONFIRMEDPDU
   other                         (0), -- OTHER
   unrecognized-service          (1), -- UNRECOGNIZED-SERVICE
```

```
    invalid-argument              (2), -- INVALID-ARGUMENT
    max-recursion-exceeded        (3), -- MAX-RECURSION-EXCEEDED
    value-out-of-range            (4) -- VALUE-OUT-OF-RANGE
  },
  pdu-error                     [5] IMPLICIT INTEGER {-- PDU-ERROR
    unknown-pdu-type              (0), -- UNKNOWN-PDU-TYPE
    invalid-pdu                   (1), -- INVALID-PDU
    illegal-acse-mapping          (2) -- ILLEGAL-ACSE-MAPPING
  },
IF ( cancel )
  cancel-requestPDU             [6] IMPLICIT INTEGER { -- CANCEL-REQUESTPDU
    other                         (0), -- OTHER
    invalid-invokeID              (1) -- INVALID-INVOKEID
  },
  cancel-responsePDU            [7] IMPLICIT INTEGER { -- CANCEL-RESPONSEPDU
    other                         (0), -- OTHER
    invalid-invokeID              (1) -- INVALID-INVOKEID
  },
  cancel-errorPDU               [8] IMPLICIT INTEGER { -- CANCEL-ERRORPDU
    other                         (0), -- OTHER
    invalid-invokeID              (1), -- INVALID-INVOKEID
    invalid-serviceError          (2), -- INVALID-SERVICEERROR
    value-out-of-range            (3) -- VALUE-OUT-OF-RANGE
  },
ENDIF
  conclude-requestPDU           [9] IMPLICIT INTEGER { -- CONCLUDE-REQUESTPDU
    other                         (0), -- OTHER
    invalid-argument              (1) -- INVALID-ARGUMENT
  },
  conclude-responsePDU          [10] IMPLICIT INTEGER { -- CONCLUDE-RESPONSEPDU
    other                         (0), -- OTHER
    invalid-result                (1) -- INVALID-RESULT
  },
  conclude-errorPDU             [11] IMPLICIT INTEGER { -- CONCLUDE-ERRORPDU
    other                         (0), -- OTHER
    invalid-serviceError          (1), -- INVALID-SERVICEERROR
    value-out-of-range            (2) -- VALUE-OUT-OF-RANGE
  } } }
```

4.32 Status

```
Status-Request ::= BOOLEAN -- Extended Derivation

Status-Response ::= SEQUENCE {
  vmdLogicalStatus          [0] IMPLICIT INTEGER {
  state-changes-allowed           (0),
  no-state-changes-allowed        (1),
  limited-services-permitted      (2),
  support-services-allowed        (3) },
```

```
 vmdPhysicalStatus          [1] IMPLICIT INTEGER {
 operational                      (0),
 partially-operational            (1),
 inoperable                       (2),
 needs-commissioning              (3) },
 localDetail                [2] IMPLICIT BIT STRING (SIZE(0..128)) OPTIONAL
            -- not to exceed 128 bits in length
 }

CS-Status-Response ::=
IF ( csr )
RobotStatusDetail ::= SEQUENCE {
 robotVMDState              [0] IMPLICIT RobotVMDState,
 robotSpecificStatus        [1] IMPLICIT RobotSpecificStatus,
 robotSpecificStatusMask    [2] IMPLICIT RobotSpecificStatus DEFAULT '1111'B,
 selectedProgramInvocation CHOICE {
 programInvocation          [3] IMPLICIT Identifier,
 noneSelected               [4] IMPLICIT NULL    } }
ENDIF
IF ( cspi )
 NULL
ENDIF
```

4.33 Robot Status

```
RobotVMDState ::= INTEGER {
 robot-idle                    (0),
 robot-loaded                  (1),
 robot-ready                   (2),
 robot-executing               (3),
 robot-motion-paused           (4),
 manualInterventionRequired    (5) }

RobotSpecificStatus ::= BITSTRING {
 safetyInterlocksViolated        (0),
 anyPhysicalResourcePowerOn      (1),
 allPhysicalResourcesCalibrated  (2),
 localControl                    (3) }
```

4.34 UnsolicitedStatus

```
UnsolicitedStatus ::= Status-Response
```

4.35 GetNameList

```
GetNameList-Request ::= SEQUENCE {
 extendedObjectClass     [0] CHOICE {
 objectClass                [0] IMPLICIT INTEGER {
IF ( vnam )
   namedVariable               (0),
ENDIF
IF ( vsca )
   scatteredAccess             (1),
ENDIF
IF ( vlis )
   namedVariableList           (2),
ENDIF
IF ( vnam )
   namedType                   (3),
ENDIF
   semaphore                   (4),
   eventCondition              (5),
   eventAction                 (6),
   eventEnrollment             (7),
   journal                     (8),
   domain                      (9),
   programInvocation           (10),
   operatorStation             (11),
   dataExchange                (12),
      -- Shall not appear in minor version 1
   accessControlList           (13) },
      -- Shall not appear in minor version 1 or y
IF ( cspi )
 csObjectClass               [1] CsAdditionalObjectClasses
ENDIF
 },
 objectScope               [1] CHOICE {
 vmdSpecific                  [0] IMPLICIT NULL,
 domainSpecific               [1] IMPLICIT Identifier,
 aaSpecific                   [2] IMPLICIT NULL },
 continueAfter             [2] IMPLICIT Identifier OPTIONAL }

GetNameList-Response ::= SEQUENCE {
 listOfIdentifier    [0] IMPLICIT SEQUENCE OF Identifier,
 moreFollows         [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.36 CsAdditionalObjectClasses

```
CsAdditionalObjectClasses ::= IMPLICIT INTEGER {
 eventConditionList        (0),
 unitControl               (1) }
```

4.37 Identify

```
Identify-Request ::= NULL

Identify-Response ::= SEQUENCE {
 vendorName              [0] IMPLICIT VisibleString,
 modelName               [1] IMPLICIT VisibleString,
 revision                [2] IMPLICIT VisibleString,
 listOfAbstractSyntaxes [3] IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL }
```

4.38 Rename

```
Rename-Request ::= SEQUENCE {
 extendedObjectClass    [0] CHOICE {
 objectClass               [0] IMPLICIT INTEGER {
IF ( vnam )
   namedVariable              (0),
ENDIF
IF ( vsca )
   scatteredAccess            (1),
ENDIF
IF ( vlis )
   namedVariableList          (2),
ENDIF
IF ( vnam )
   namedType                  (3),
ENDIF
   semaphore                  (4),
   eventCondition             (5),
   eventAction                (6),
   eventEnrollment            (7),
   journal                    (8),
   domain                     (9),
   programInvocation          (10),
   operatorStation            (11),
   dataExchange               (12),
      -- Shall not appear in minor version 1
   accessControlList          (13) },
      -- Shall not appear in minor version 1 or y
IF ( cspi )
 csObjectClass             [1] CsAdditionalObjectClasses
ENDIF
```

```
 },
 currentName              [1] ObjectName,
 newIdentifier            [2] IMPLICIT Identifier }

Rename-Response ::= NULL
```

4.39 GetCapabilityList

```
GetCapabilityList-Request ::= SEQUENCE {
 continueAfter           VisibleString OPTIONAL }

GetCapabilityList-Response ::= SEQUENCE {
 listOfCapabilities  [0] IMPLICIT SEQUENCE OF VisibleString,
 moreFollows         [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.40 InitiateDownloadSequence

```
InitiateDownloadSequence-Request ::= SEQUENCE {
 domainName          [0] IMPLICIT Identifier,
 listOfCapabilities  [1] IMPLICIT SEQUENCE OF VisibleString,
 sharable            [2] IMPLICIT BOOLEAN }

InitiateDownloadSequence-Response ::= NULL
```

4.41 DownloadSegment

```
DownloadSegment-Request ::= Identifier -- Domain Name

DownloadSegment-Response ::= SEQUENCE {
 loadData            CHOICE {
 non-coded              [0] IMPLICIT OCTET STRING,
 coded                  EXTERNAL },
 moreFollows         [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.42 TerminateDownloadSequence

```
TerminateDownloadSequence-Request ::= SEQUENCE {
 domainName          [0] IMPLICIT Identifier,
 discard             [1] IMPLICIT ServiceError OPTIONAL }

TerminateDownloadSequence-Response ::= NULL
```

4.43 InitiateUploadSequence

```
InitiateUploadSequence-Request ::= Identifier -- Domain Name

InitiateUploadSequence-Response ::= SEQUENCE {
 ulsmID               [0] IMPLICIT Integer32,
 listOfCapabilities   [1] IMPLICIT SEQUENCE OF VisibleString }
```

4.44 UploadSegment

```
UploadSegment-Request ::= Integer32 -- ULSM ID

UploadSegment-Response ::= SEQUENCE {
 loadData             CHOICE {
 non-coded               [0] IMPLICIT OCTET STRING,
 coded                   EXTERNAL },
 moreFollows          [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.45 TerminateUploadSequence

```
TerminateUploadSequence-Request ::= Integer32 -- ULSM ID

TerminateUploadSequence-Response ::= NULL
```

4.46 RequestDomainDownload

```
RequestDomainDownload-Request ::= SEQUENCE {
 domainName           [0] IMPLICIT Identifier,
 listOfCapabilities   [1] IMPLICIT SEQUENCE OF VisibleString OPTIONAL,
 sharable             [2] IMPLICIT BOOLEAN,
 fileName             [4] IMPLICIT FileName }

RequestDomainDownload-Response ::= NULL
```

4.47 RequestDomainUpload

```
RequestDomainUpload-Request ::= SEQUENCE {
 domainName           [0] IMPLICIT Identifier,
 fileName             [1] IMPLICIT FileName }

RequestDomainUpload-Response ::= NULL
```

4.48 LoadDomainContent

```
LoadDomainContent-Request ::= SEQUENCE {
 domainName           [0] IMPLICIT Identifier,
 listOfCapabilities   [1] IMPLICIT SEQUENCE OF VisibleString OPTIONAL,
 sharable             [2] IMPLICIT BOOLEAN,
 fileName             [4] IMPLICIT FileName,
IF ( tpy )
 thirdParty           [5] IMPLICIT ApplicationReference OPTIONAL
ENDIF
 }

LoadDomainContent-Response ::= NULL
```

4.49 StoreDomainContent

```
StoreDomainContent-Request ::= SEQUENCE {
 domainName           [0] IMPLICIT Identifier,
 fileName             [1] IMPLICIT FileName,
IF ( tpy )
 thirdParty           [2] IMPLICIT ApplicationReference OPTIONAL
ENDIF
 }

StoreDomainContent-Response ::= NULL
```

4.50 DeleteDomain

```
DeleteDomain-Request ::= Identifier -- Domain Name

DeleteDomain-Response ::= NULL
```

4.51 GetDomainAttributes

```
GetDomainAttributes-Request ::= Identifier -- Domain Name

GetDomainAttributes-Response ::= SEQUENCE {
 listOfCapabilities        [0] IMPLICIT SEQUENCE OF VisibleString,
 state                     [1] IMPLICIT DomainState,
 mmsDeletable              [2] IMPLICIT BOOLEAN,
 sharable                  [3] IMPLICIT BOOLEAN,
 listOfProgramInvocations  [4] IMPLICIT SEQUENCE OF Identifier,
            -- Program Invocation Names
 uploadInProgress          [5] IMPLICIT Integer8,
 accessControlList         [6] IMPLICIT Identifier OPTIONAL }
```

4.52 DomainState

```
DomainState ::= INTEGER {
 non-existent      (0),   -- NON-EXISTENT
                          -- shall not be reported
 loading           (1),   -- LOADED
 ready             (2),   -- READY
 in-use            (3),   -- IN-USE
 complete          (4),   -- COMPLETE
 incomplete        (5),   -- INCOMPLETE
 d1                (7),   -- D1
 d2                (8),   -- D2
 d3                (9),   -- D3
 d4                (10),  -- D4
 d5                (11),  -- D5
 d6                (12),  -- D6
 d7                (13),  -- D7
 d8                (14),  -- D8
 d9                (15)   -- D9
 }
```

4.53 CreateProgramInvocation

```
CreateProgramInvocation-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier,
 listOfDomainNames      [1] IMPLICIT SEQUENCE OF Identifier,
 reusable               [2] IMPLICIT BOOLEAN DEFAULT TRUE,
IF ( eventNotification )
IF ( getProgramInvocationAttributes )
 monitorType            [3] IMPLICIT BOOLEAN OPTIONAL
      -- TRUE indicates PERMANENT monitoring,
      -- FALSE indicates CURRENT monitoring
ENDIF
ENDIF
 }

CreateProgramInvocation-Response ::= NULL

CS-CreateProgramInvocation-Request ::= INTEGER {
 normal            (0),
 controlling       (1),
 controlled        (2) }
```

4.54 DeleteProgramInvocation

```
DeleteProgramInvocation-Request ::= Identifier -- Program Invocation Name

DeleteProgramInvocation-Response ::= NULL
```

4.55 Start

```
Start-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier,
 executionArgument      CHOICE {
 simpleString              [1] IMPLICIT VisibleString,
 encodedString             EXTERNAL } OPTIONAL }

CS-Start-Request ::= [0] CHOICE {
 normal            NULL,
 controlling       SEQUENCE {
 startLocation        [0] IMPLICIT VisibleString OPTIONAL,
 startCount           [1] StartCount DEFAULT cycleCount 1 } }

Start-Response ::= NULL

StartCount ::= CHOICE {
 noLimit        [0] IMPLICIT NULL,
 cycleCount     [1] IMPLICIT INTEGER,
 stepCount      [2] IMPLICIT INTEGER }

Start-Error::= ProgramInvocationState
```

4.56 ProgramInvocationState

```
ProgramInvocationState ::= INTEGER {
 non-existent          (0),   -- NON-EXISTENT
                              -- shall not be reported
 unrunnable            (1),   -- UNRUNNABLE
 idle                  (2),   -- IDLE
 running               (3),   -- RUNNING
 stopped               (4),   -- STOPPED
 starting              (5),   -- STARTING
 stopping              (6),   -- STOPPING
 resuming              (7),   -- RESUMING
 resetting             (8)    -- RESETTING
 }
```

4.57 Stop

```
Stop-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier }

Stop-Response ::= NULL

Stop-Error ::= ProgramInvocationState
```

4.58 Resume

```
Resume-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier,
 executionArgument      CHOICE {
 simpleString              [1] IMPLICIT VisibleString,
 encodedString             EXTERNAL } OPTIONAL }

Resume-Response ::= NULL

CS-Resume-Request ::= [0] CHOICE {
 normal               NULL,
 controlling          SEQUENCE { CHOICE {
 continueMode                    [0] IMPLICIT NULL,
 changeMode                      [1] StartCount } } }

Resume-Error    ::= ProgramInvocationState
```

4.59 Reset

```
Reset-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier }

Reset-Response ::= NULL

Reset-Error ::= ProgramInvocationState
```

4.60 Kill

```
Kill-Request ::= SEQUENCE {
 programInvocationName  [0] IMPLICIT Identifier }

Kill-Response ::= NULL
```

4.61 GetProgramInvocationAttributes

```
GetProgramInvocationAttributes-Request ::= Identifier -- Program Invocation Name

GetProgramInvocationAttributes-Response ::= SEQUENCE {
 state                [0] IMPLICIT ProgramInvocationState,
 listOfDomainNames    [1] IMPLICIT SEQUENCE OF Identifier,
 mmsDeletable         [2] IMPLICIT BOOLEAN,
 reusable             [3] IMPLICIT BOOLEAN,
 monitor              [4] IMPLICIT BOOLEAN,
 executionArgument    CHOICE {
   simpleString          [5] IMPLICIT VisibleString,
   encodedString         EXTERNAL
   },
 accessControlList        [6] IMPLICIT Identifier OPTIONAL }


CS-GetProgramInvocationAttributes-Response ::= SEQUENCE {
 errorCode            [0] IMPLICIT INTEGER,
 control              [1] CHOICE {
 controlling             [0] IMPLICIT SEQUENCE {
   controlledPI             [0] IMPLICIT SEQUENCE OF Identifier,
   programLocation          [1] IMPLICIT VisibleString OPTIONAL,
   runningMode              [2] CHOICE {
   freeRunning                 [0] IMPLICIT NULL,
   cycleLimited                [1] IMPLICIT INTEGER,
   stepLimited                 [2] IMPLICIT INTEGER } },
 controlled              [1] CHOICE {
   controllingPI            [0] IMPLICIT Identifier,
                         -- Reference to Controlling
   none                     [1] IMPLICIT NULL -- uncontrolled
   },
 normal               [2] IMPLICIT NULL } }
```

4.62 TypeSpecification

```
TypeSpecification ::= CHOICE {
IF ( vnam )
 typeName          [0] ObjectName,
ENDIF
IF ( str1 )
 array             [1] IMPLICIT SEQUENCE {
 packed               [0] IMPLICIT BOOLEAN DEFAULT FALSE,
 numberOfElements         [1] IMPLICIT Unsigned32,
 elementType              [2] TypeSpecification },
ENDIF
IF ( str2 )
 structure         [2] IMPLICIT SEQUENCE {
```

```
 packed                [0] IMPLICIT BOOLEAN DEFAULT FALSE,
 components            [1] IMPLICIT SEQUENCE OF SEQUENCE {
   componentName             [0] IMPLICIT Identifier OPTIONAL,
   componentType             [1] TypeSpecification } },
ENDIF
-- Simple Size Class
 boolean           [3] IMPLICIT NULL,       -- BOOLEAN
 bit-string        [4] IMPLICIT Integer32,  -- BIT-STRING
 integer           [5] IMPLICIT Unsigned8,  -- INTEGER
 unsigned          [6] IMPLICIT Unsigned8,  -- UNSIGNED
 floating-point    [7] IMPLICIT SEQUENCE {
 format-width          Unsigned8, -- number of bits in fraction plus sign
 exponent-width        Unsigned8 -- size of exponent in bits
 },
IF ( real )
 real              [8] IMPLICIT SEQUENCE {
 base                  [0] IMPLICIT INTEGER(2|10),
 exponent              [1] IMPLICIT INTEGER, -- max number of octets
 mantissa              [2] IMPLICIT INTEGER -- max number of octets
 },
ENDIF
 octet-string      [9] IMPLICIT Integer32, -- OCTET-STRING
 visible-string    [10] IMPLICIT Integer32,   -- VISIBLE-STRING
 generalized-time  [11] IMPLICIT NULL,     -- GENERALIZED-TIME
 binary-time       [12] IMPLICIT BOOLEAN,  -- BINARY-TIME
 bcd               [13] IMPLICIT Unsigned8,    -- BCD
 objId             [15] IMPLICIT NULL }
```

4.63 AlternateAccess

```
AlternateAccess ::= SEQUENCE OF CHOICE {
 unnamed           AlternateAccessSelection,
IF ( str2 )
 named             [5] IMPLICIT SEQUENCE {
 componentName         [0] IMPLICIT Identifier,
 access                AlternateAccessSelection }
ENDIF
 }

AlternateAccessSelection ::= CHOICE {
 selectAlternateAccess  [0] IMPLICIT SEQUENCE {
 accessSelection            CHOICE {
IF ( str2 )
 component                     [0] IMPLICIT Identifier,    -- component
ENDIF
IF ( str1 )
 index                         [1] IMPLICIT Unsigned32,    -- 1 array element
 indexRange                    [2] IMPLICIT SEQUENCE {-- array elements
   lowIndex                       [0] IMPLICIT Unsigned32,
```

```
   numberOfElements                [1] IMPLICIT Unsigned32 },
ENDIF
 allElements                     [3] IMPLICIT NULL -- all array elements
 },
 alternateAccess                 AlternateAccess },
 selectAccess                  CHOICE {
IF ( str2 )
 component                       [1] IMPLICIT Identifier, -- component
ENDIF
IF ( str1 )
 index                           [2] IMPLICIT Unsigned32,    -- 1 array element
 indexRange                      [3] IMPLICIT SEQUENCE {-- array elements
   lowIndex                        [0] IMPLICIT Unsigned32,
   numberOfElements                [1] IMPLICIT Unsigned32 },
ENDIF
 allElements                     [4] IMPLICIT NULL -- all array elements
 } }
```

4.64 AccessResult

```
AccessResult ::= CHOICE {
 failure          [0] IMPLICIT DataAccessError,
 success          Data }
```

4.65 Data

```
Data ::= CHOICE {
      -- context tag 0 is reserved for AccessResult
IF ( str1 )
 array               [1] IMPLICIT SEQUENCE OF Data,
ENDIF
IF ( str2 )
 structure           [2] IMPLICIT SEQUENCE OF Data,
ENDIF
 boolean             [3] IMPLICIT BOOLEAN,
 bit-string          [4] IMPLICIT BIT STRING,
 integer             [5] IMPLICIT INTEGER,
 unsigned            [6] IMPLICIT INTEGER,  -- shall not be negative
 floating-point      [7] IMPLICIT FloatingPoint,
IF ( real )
 real                [8] IMPLICIT REAL,
ENDIF
 octet-string        [9] IMPLICIT OCTET STRING,
 visible-string      [10] IMPLICIT VisibleString,
 generalized-time    [11] IMPLICIT GeneralizedTime,
 binary-time         [12] IMPLICIT TimeOfDay,
 bcd                 [13] IMPLICIT INTEGER, -- shall not be negative
 booleanArray        [14] IMPLICIT BIT STRING,
 objId               [15] IMPLICIT OBJECT IDENTIFIER }
```

```
FloatingPoint ::= OCTET STRING
```

4.66 DataAccessError

```
DataAccessError ::= INTEGER {
 object-invalidated              (0),      -- OBJECT-INVALIDATED
 hardware-fault                  (1),      -- HARDWARE-FAULT
 temporarily-unavailable         (2),      -- TEMPORARILY-UNAVAILABLE
 object-access-denied            (3),      -- OBJECT-ACCESS-DENIED
 object-undefined                (4),      -- OBJECT-UNDEFINED
 invalid-address                 (5),      -- INVALID-ADDRESS
 type-unsupported                (6),      -- TYPE-UNSUPPORTED
 type-inconsistent               (7),      -- TYPE-INCONSISTENT
 object-attribute-inconsistent   (8),      -- OBJECT-ATTRIBUTE-INCONSISTENT
 object-access-unsupported       (9),      -- OBJECT-ACCESS-UNSUPPORTED
 object-non-existent             (10),     -- OBJECT-NON-EXISTENT
 object-value-invalid            (11)      -- OBJECT-VALUE-INVALID
 }
```

4.67 VariableAccessSpecification

```
VariableAccessSpecification ::= CHOICE {
 listOfVariable      [0] IMPLICIT SEQUENCE OF SEQUENCE {
 variableSpecification     VariableSpecification,
IF ( valt )
 alternateAccess           [5] IMPLICIT AlternateAccess OPTIONAL,
ENDIF
 }IF ( vlis )
 variableListName    [1] ObjectName
ENDIF
 }
```

4.68 VariableSpecification

```
VariableSpecification ::= CHOICE {
IF ( vnam )
 name                          [0] ObjectName,
ENDIF
IF ( vadr )
 address                       [1] Address,
 variableDescription           [2] IMPLICIT SEQUENCE {
 address                          Address,
 typeSpecification                TypeSpecification },
ENDIF
IF ( vsca )
 scatteredAccessDescription    [3] IMPLICIT ScatteredAccessDescription,
ENDIF
 invalidated                   [4] IMPLICIT NULL }
```

4.69 ScatteredAccessDescription

```
ScatteredAccessDescription ::= SEQUENCE OF SEQUENCE {
 componentName              [0] IMPLICIT Identifier OPTIONAL,
 variableSpecification      [1] VariableSpecification.
IF ( valt )
 alternateAccess            [2] IMPLICIT AlternateAccess OPTIONAL
ENDIF
 }
```

4.70 Address

```
Address ::= CHOICE {
 numericAddress             [0] IMPLICIT Unsigned32,
 symbolicAddress            [1] IMPLICIT VisibleString,
 unconstrainedAddress       [2] IMPLICIT OCTET STRING }
```

4.71 Read

```
Read-Request ::= SEQUENCE {
 specificationWithResult       [0] IMPLICIT BOOLEAN DEFAULT FALSE,
 variableAccessSpecification   [1] VariableAccessSpecification }

Read-Response ::= SEQUENCE {
 variableAccessSpecification   [0] VariableAccessSpecification OPTIONAL,
 listOfAccessResult            [1] IMPLICIT SEQUENCE OF AccessResult }
```

4.72 Write

```
Write-Request ::= SEQUENCE {
 variableAccessSpecification  VariableAccessSpecification,
 listOfData                   [0] IMPLICIT SEQUENCE OF Data }

Write-Response ::= SEQUENCE OF CHOICE {
 failure                [0] IMPLICIT DataAccessError,
 success                [1] IMPLICIT NULL }
```

4.73 InformationReport

```
InformationReport ::= SEQUENCE {
 variableAccessSpecification  VariableAccessSpecification,
 listOfAccessResult           [0] IMPLICIT SEQUENCE OF AccessResult }
```

4.74 GetVariableAccessAttributes

```
GetVariableAccessAttributes-Request ::= CHOICE {
IF ( vnam )
 name                    [0] ObjectName,
ENDIF
IF ( vadr )
 address                 [1] Address
ENDIF
 }

GetVariableAccessAttributes-Response ::= SEQUENCE {
 mmsDeletable            [0] IMPLICIT BOOLEAN,
IF ( vadr )
 address                 [1] Address OPTIONAL,
ENDIF
 typeSpecification       [2] TypeSpecification,
 accessControlList       [3] IMPLICIT Identifier OPTIONAL }
```

4.75 DefineNamedVariable

```
DefineNamedVariable-Request ::= SEQUENCE {
 variableName            [0] ObjectName,
 address                 [1] Address,
 typeSpecification       [2] TypeSpecification OPTIONAL }

DefineNamedVariable-Response ::= NULL
```

4.76 DefineScatteredAccess

```
DefineScatteredAccess-Request ::= SEQUENCE {
 scatteredAccessName          [0] ObjectName,
 scatteredAccessDescription   [1] IMPLICIT ScatteredAccessDescription }

DefineScatteredAccess-Response ::= NULL
```

4.77 GetScatteredAccessAttributes

```
GetScatteredAccessAttributes-Request ::= ObjectName --ScatteredAccessName

GetScatteredAccessAttributes-Response ::= SEQUENCE {
 mmsDeletable                 [0] IMPLICIT BOOLEAN,
 scatteredAccessDescription   [1] IMPLICIT ScatteredAccessDescription,
 accessControlList            [2] IMPLICIT Identifier OPTIONAL }
```

4.78 DeleteVariableAccess

```
DeleteVariableAccess-Request ::= SEQUENCE {
 scopeOfDelete       [0] IMPLICIT INTEGER {
 specific                  (0), -- SPECIFIC
 aa-specific               (1), -- AA-SPECIFIC
 domain                    (2), -- DOMAIN
 vmd                       (3) -- VMD
 } DEFAULT specific,
 listOfName          [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
 domainName          [2] IMPLICIT Identifier OPTIONAL }

DeleteVariableAccess-Response ::= SEQUENCE {
 numberMatched       [0] IMPLICIT Unsigned32,
 numberDeleted       [1] IMPLICIT Unsigned32 }

DeleteVariableAccess-Error ::= Unsigned32 -- numberDeleted
```

4.79 DefineNamedVariableList

```
DefineNamedVariableList-Request ::= SEQUENCE {
 variableListName    ObjectName,
 listOfVariable      [0] IMPLICIT SEQUENCE OF SEQUENCE {
 variableSpecification     VariableSpecification,
IF ( valt )
 alternateAccess           [5] IMPLICIT AlternateAccess OPTIONAL
ENDIF
 } }

DefineNamedVariableList-Response ::= NULL
```

4.80 GetNamedVariableListAttributes

```
GetNamedVariableListAttributes-Request ::= ObjectName -- VariableListName

GetNamedVariableListAttributes-Response ::= SEQUENCE {
 mmsDeletable        [0] IMPLICIT BOOLEAN,
 listOfVariable      [1] IMPLICIT SEQUENCE OF SEQUENCE {
   variableSpecification   VariableSpecification,
IF ( valt )
   alternateAccess         [5] IMPLICIT AlternateAccess OPTIONAL
ENDIF
      },
 accessControlList   [2] IMPLICIT Identifier OPTIONAL }
```

4.81 DeleteNamedVariableList

```
DeleteNamedVariableList-Request ::= SEQUENCE {
 scopeOfDelete          [0] IMPLICIT INTEGER {
 specific                  (0), -- SPECIFIC
 aa-specific               (1), -- AA-SPECIFIC
 domain                    (2), -- DOMAIN
 vmd                       (3) -- VMD
 } DEFAULT specific,
 listOfVariableListName [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
 domainName             [2] IMPLICIT Identifier OPTIONAL }

DeleteNamedVariableList-Response ::= SEQUENCE {
 numberMatched       [0] IMPLICIT Unsigned32,
 numberDeleted       [1] IMPLICIT Unsigned32 }

DeleteNamedVariableList-Error ::= Unsigned32 -- numberDeleted
```

4.82 DefineNamedType

```
DefineNamedType-Request ::= SEQUENCE {
 typeName               ObjectName,
 typeSpecification      TypeSpecification }

DefineNamedType-Response ::= NULL
```

4.83 GetNamedTypeAttributes

```
GetNamedTypeAttributes-Request ::= ObjectName --TypeName

GetNamedTypeAttributes-Response ::= SEQUENCE {
 mmsDeletable           [0] IMPLICIT BOOLEAN,
 typeSpecification      TypeSpecification,
 accessControlList      Identifier OPTIONAL
 }
```

4.84 DeleteNamedType

```
DeleteNamedType-Request ::= SEQUENCE {
 scopeOfDelete       [0] IMPLICIT INTEGER {
 specific               (0), -- SPECIFIC
 aa-specific            (1), -- AA-SPECIFIC
 domain                 (2), -- DOMAIN
 vmd                    (3) -- VMD
 } DEFAULT specific,
 listOfTypeName      [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
 domainName          [2] IMPLICIT Identifier OPTIONAL }
```

```
DeleteNamedType-Response ::= SEQUENCE {
 numberMatched              [0] IMPLICIT Unsigned32,
 numberDeleted              [1] IMPLICIT Unsigned32 }

DeleteNamedType-Error ::= Unsigned32 -- numberDeleted
```

4.85 TakeControl

```
TakeControl-Request ::= SEQUENCE {
 semaphoreName                [0] ObjectName,
 namedToken                   [1] IMPLICIT Identifier OPTIONAL,
 priority                     [2] IMPLICIT Priority DEFAULT normalPriority,
 acceptableDelay              [3] IMPLICIT Unsigned32 OPTIONAL,
 controlTimeOut               [4] IMPLICIT Unsigned32 OPTIONAL,
 abortOnTimeOut               [5] IMPLICIT BOOLEAN OPTIONAL,
 relinquishIfConnectionLost   [6] IMPLICIT BOOLEAN DEFAULT TRUE,
IF ( tpy )
 applicationToPreempt         [7] IMPLICIT ApplicationReference OPTIONAL
ENDIF
 }

TakeControl-Response ::= CHOICE {
 noResult            [0] IMPLICIT NULL,
 namedToken          [1] IMPLICIT Identifier }
```

4.86 RelinquishControl

```
RelinquishControl-Request ::= SEQUENCE {
 semaphoreName       [0] ObjectName,
 namedToken          [1] IMPLICIT Identifier OPTIONAL }

RelinquishControl-Response ::= NULL
```

4.87 DefineSemaphore

```
DefineSemaphore-Request ::= SEQUENCE {
 semaphoreName       [0] ObjectName,
 numberOfTokens      [1] IMPLICIT Unsigned16 }

DefineSemaphore-Response ::= NULL
```

4.88 DeleteSemaphore

```
DeleteSemaphore-Request ::= ObjectName -- Semaphore Name

DeleteSemaphore-Response ::= NULL
```

4.89 ReportSemaphoreStatus

```
ReportSemaphoreStatus-Request ::= ObjectName -- Semaphore Name

ReportSemaphoreStatus-Response ::= SEQUENCE {
 mmsDeletable         [0] IMPLICIT BOOLEAN,
 class                [1] IMPLICIT INTEGER {
 token                   (0),
 pool                    (1) },
 numberOfTokens       [2] IMPLICIT Unsigned16,
 numberOfOwnedTokens  [3] IMPLICIT Unsigned16,
 numberOfHungTokens   [4] IMPLICIT Unsigned16,
 accessControlList    [5] IMPLICIT Identifier OPTIONAL }
```

4.90 ReportPoolSemaphoreStatus

```
ReportPoolSemaphoreStatus-Request ::= SEQUENCE {
 semaphoreName        [0] ObjectName,
 nameToStartAfter     [1] IMPLICIT Identifier OPTIONAL }

ReportPoolSemaphoreStatus-Response ::= SEQUENCE {
 listOfNamedTokens    [0] IMPLICIT SEQUENCE OF CHOICE {
 freeNamedToken           [0] IMPLICIT Identifier,
 ownedNamedToken          [1] IMPLICIT Identifier,
 hungNamedToken           [2] IMPLICIT Identifier },
 moreFollows          [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.91 ReportSemaphoreEntryStatus

```
ReportSemaphoreEntryStatus-Request ::= SEQUENCE {
 semaphoreName        [0] ObjectName,
 state                [1] IMPLICIT INTEGER {
 queued                  (0),
 owner                   (1),
 hung                    (2) },
 entryIDToStartAfter  [2] IMPLICIT OCTET STRING OPTIONAL }
```

```
ReportSemaphoreEntryStatus-Response ::= SEQUENCE {
 listOfSemaphoreEntry    [0] IMPLICIT SEQUENCE OF SemaphoreEntry,
 moreFollows             [1] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.92 AttachToSemaphore

```
AttachToSemaphore ::= SEQUENCE {
 semaphoreName                  [0] ObjectName,
 namedToken                     [1] IMPLICIT Identifier OPTIONAL,
 priority                       [2] IMPLICIT Priority DEFAULT normalPriority,
 acceptableDelay                [3] IMPLICIT Unsigned32 OPTIONAL,
 controlTimeOut                 [4] IMPLICIT Unsigned32 OPTIONAL,
 abortOnTimeOut                 [5] IMPLICIT BOOLEAN OPTIONAL,
 relinquishIfConnectionLost     [6] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.93 SemaphoreEntry

```
SemaphoreEntry ::= SEQUENCE {
 entryID                        [0] IMPLICIT OCTET STRING,
 entryClass                     [1] IMPLICIT INTEGER {
 simple                             (0),
 modifier                           (1) },
 applicationReference           [2] ApplicationReference,
 namedToken                     [3] IMPLICIT Identifier OPTIONAL,
 priority                       [4] IMPLICIT Priority DEFAULT normalPriority,
 remainingTimeOut               [5] IMPLICIT Unsigned32 OPTIONAL,
 abortOnTimeOut                 [6] IMPLICIT BOOLEAN OPTIONAL,
 relinquishIfConnectionLost     [7] IMPLICIT BOOLEAN DEFAULT TRUE }
```

4.94 Input

```
Input-Request ::= SEQUENCE {
 operatorStationName [0] IMPLICIT Identifier,
 echo                [1] IMPLICIT BOOLEAN DEFAULT TRUE,
 listOfPromptData    [2] IMPLICIT SEQUENCE OF VisibleString OPTIONAL,
 inputTimeOut        [3] IMPLICIT Unsigned32 OPTIONAL }

Input-Response ::= VisibleString -- Input String
```

4.95 Output

```
Output-Request ::= SEQUENCE {
 operatorStationName [0] IMPLICIT Identifier,
 listOfOutputData    [1] IMPLICIT SEQUENCE OF VisibleString }
```

```
Output-Response ::= NULL
```

4.96 DefineEventCondition

```
DefineEventCondition-Request ::= SEQUENCE {
 eventConditionName  [0] ObjectName,
 class               [1] IMPLICIT EC-Class,
 priority            [2] IMPLICIT Priority DEFAULT normalPriority,
 severity            [3] IMPLICIT Unsigned8 DEFAULT normalSeverity,
 alarmSummaryReports [4] IMPLICIT BOOLEAN OPTIONAL,
 monitoredVariable   [6] VariableSpecification OPTIONAL,
 evaluationInterval  [7] IMPLICIT Unsigned32 OPTIONAL }

DefineEventCondition-Response ::= NULL

CS-DefineEventCondition-Request ::= [0] CHOICE {
IF ( des )
 enhancementString      [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex       [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement          NULL }
```

4.97 DeleteEventCondition

```
DeleteEventCondition-Request ::= CHOICE {
 specific            [0] IMPLICIT SEQUENCE OF ObjectName,
 aa-specific         [1] IMPLICIT NULL,
 domain              [2] IMPLICIT Identifier,
 vmd                 [3] IMPLICIT NULL }

DeleteEventCondition-Response ::= Unsigned32 --Candidates Not Deleted
```

4.98 GetEventConditionAttributes

```
GetEventConditionAttributes-Request ::= ObjectName --Event Condition Name

GetEventConditionAttributes-Response ::= SEQUENCE {
 mmsDeletable        [0] IMPLICIT BOOLEAN DEFAULT FALSE,
 class               [1] IMPLICIT EC-Class,
 priority            [2] IMPLICIT Priority DEFAULT normalPriority,
 severity            [3] IMPLICIT Unsigned8 DEFAULT normalSeverity,
 alarmSummaryReports [4] IMPLICIT BOOLEAN DEFAULT FALSE,
 monitoredVariable   [6] CHOICE {
 variableReference      [0] VariableSpecification,
 undefined              [1] IMPLICIT NULL } OPTIONAL,
 evaluationInterval  [7] IMPLICIT Unsigned32 OPTIONAL,
```

```
 accessControlList    [8] IMPLICIT Identifier OPTIONAL }

CS-GetEventConditionAttributes-Response ::= SEQUENCE {
 groupPriorityOverride      [0] CHOICE {
 priority                      [0] IMPLICIT Priority,
 undefined                     [1] IMPLICIT NULL } OPTIONAL,
 listOfEventConditionList   [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
 displayEnhancement         [2] CHOICE {
IF ( des )
 enhancementString             [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex              [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement                 [2] IMPLICIT NULL } }
```

4.99 ReportEventConditionStatus

```
ReportEventConditionStatus-Request ::= ObjectName --Event Condition Name

ReportEventConditionStatus-Response ::= SEQUENCE {
 currentState                  [0] IMPLICIT EC-State,
 numberOfEventEnrollments      [1] IMPLICIT Unsigned32,
 enabled                       [2] IMPLICIT BOOLEAN OPTIONAL,
 timeOfLastTransitionToActive [3] EventTime OPTIONAL,
 timeOfLastTransitionToIdle    [4] EventTime OPTIONAL }
```

4.100 AlterEventConditionMonitoring

```
AlterEventConditionMonitoring-Request ::= SEQUENCE {
 eventConditionName  [0] ObjectName,
 enabled             [1] IMPLICIT BOOLEAN OPTIONAL,
 priority            [2] IMPLICIT Priority OPTIONAL,
 alarmSummaryReports [3] IMPLICIT BOOLEAN OPTIONAL,
IF ( cei )
 evaluationInterval  [4] IMPLICIT Unsigned32 OPTIONAL
ENDIF
 }

AlterEventConditionMonitoring-Response ::= NULL

CS-AlterEventConditionMonitoring-Request ::= SEQUENCE {
 changeDisplay CHOICE {
IF ( des )
 enhancementString   [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex    [1] IMPLICIT INTEGER,
```

```
ENDIF
 noEnhancement          [2] NULL } OPTIONAL }
```

4.101 TriggerEvent

```
TriggerEvent-Request ::= SEQUENCE {
 eventConditionName  [0] ObjectName,
 priority            [1] IMPLICIT Priority OPTIONAL }

TriggerEvent-Response ::= NULL
```

4.102 DefineEventAction

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

DefineEventAction-Response ::= NULL
```

4.103 DeleteEventAction

```
DeleteEventAction-Request ::= CHOICE {
 specific             [0] IMPLICIT SEQUENCE OF ObjectName,
 aa-specific          [1] IMPLICIT NULL,
 domain               [3] IMPLICIT Identifier,
 vmd                  [4] IMPLICIT NULL }

DeleteEventAction-Response ::= Unsigned32 --Candidates Not Deleted
```

4.104 GetEventActionAttributes

```
GetEventActionAttributes-Request ::= ObjectName --EventActionName

GetEventActionAttributes-Response ::= SEQUENCE {
 mmsDeletable               [0] IMPLICIT BOOLEAN DEFAULT FALSE,
 listOfModifier             [1] IMPLICIT SEQUENCE OF Modifier,
 confirmedServiceRequest    [2] ConfirmedServiceRequest,
IF ( csr cspi )
 cs-extension               [79] CS-Request-Detail OPTIONAL,
            -- shall not be transmitted if value is the value
```

```
            -- of a tagged type derived from NULL
ENDIF
 accessControlList          [3] IMPLICIT Identifier OPTIONAL }
```

4.105 ReportEventActionStatus

```
ReportEventActionStatus-Request ::= ObjectName -- Event Action Name

ReportEventActionStatus-Response ::= Unsigned32 -- Number of Event Enrollments
```

4.106 DefineEventEnrollment

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

CS-DefineEventEnrollment-Request ::= [0] CHOICE {
IF ( des )
 enhancementString       [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex        [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement           NULL }

DefineEventEnrollment-Response ::= NULL

CS-GetEventEnrollmentAttributes-Response ::= [0] CHOICE {
IF ( des )
 enhancementString       [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex        [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement           NULL }

DefineEventEnrollment-Error ::= ObjectName
```

4.107 DeleteEventEnrollment

```
DeleteEventEnrollment-Request ::= CHOICE {
 specific       [0] IMPLICIT SEQUENCE OF ObjectName,
 ec             [1] ObjectName,
 ea             [2] ObjectName }

DeleteEventEnrollment-Response ::= Unsigned32 --Candidates Not Deleted
```

4.108 GetEventEnrollmentAttributes

```
GetEventEnrollmentAttributes-Request ::= SEQUENCE {
 scopeOfRequest          [0] IMPLICIT INTEGER {
 specific                   (0), --SPECIFIC
 client                     (1), -- CLIENT
 ec                         (2), -- EC
 ea                         (3) -- EA
 } DEFAULT client,
 eventEnrollmentNames    [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
IF ( tpy )
 clientApplication       [2] ApplicationReference OPTIONAL,
ENDIF
 eventConditionName      [3] ObjectName OPTIONAL,
 eventActionName         [4] ObjectName OPTIONAL,
 continueAfter           [5] ObjectName OPTIONAL }

GetEventEnrollmentAttributes-Response ::= SEQUENCE {
 listOfEventEnrollment   [0] IMPLICIT SEQUENCE OF EventEnrollment,
 moreFollows             [1] IMPLICIT BOOLEAN DEFAULT FALSE }
```

4.109 EventEnrollment

```
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,
IF ( tpy )
 clientApplication                  [3] ApplicationReference OPTIONAL,
ENDIF
 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,
```

```
IF ( csr cspi )
 additionalDetail                [9] IMPLICIT EE-Additional-Detail OPTIONAL,
                -- shall not be transmitted if the value is NULL
ENDIF
   accessControlList             [11] IMPLICIT Identifier OPTIONAL }
```

4.110 ReportEventEnrollmentStatus

```
ReportEventEnrollmentStatus-Request ::= ObjectName --Event Enrollment Name

ReportEventEnrollmentStatus-Response ::= SEQUENCE {
 eventConditionTransitions    [0] IMPLICIT Transitions,
 notificationLost             [1] IMPLICIT BOOLEAN DEFAULT FALSE,
 duration                     [2] IMPLICIT EE-Duration,
 alarmAcknowledgmentRule      [3] IMPLICIT AlarmAckRule OPTIONAL,
 currentState                 [4] IMPLICIT EE-State }
```

4.111 AlterEventEnrollment

```
AlterEventEnrollment-Request ::= SEQUENCE {
 eventEnrollmentName       [0] ObjectName,
 eventConditionTransitions [1] IMPLICIT Transitions OPTIONAL,
 alarmAcknowledgmentRule   [2] IMPLICIT AlarmAckRule OPTIONAL }

AlterEventEnrollment-Response ::= SEQUENCE {
 currentState         [0] CHOICE {
 state                   [0] IMPLICIT EE-State,
 undefined               [1] IMPLICIT NULL },
 transitionTime       [1] EventTime }

CS-AlterEventEnrollment-Request ::= SEQUENCE {
 changeDisplay    CHOICE {
IF ( des )
 enhancementString [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex    [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement       [2] NULL } OPTIONAL }
```

4.112 Eventnotification

```
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,
 eventActionResult              CHOICE {
   success                         [0] IMPLICIT SEQUENCE {
                                   ConfirmedServiceResponse,

IF ( csr cspi )
                                   [79] CS-Response-Detail OPTIONAL
                  -- shall not be transmitted if value is the value
                  -- of a tagged type derived from NULL
ENDIF
            },
   failure                    [1] IMPLICIT ServiceError } } OPTIONAL }

CS-EventNotification ::= [0] CHOICE {
IF ( des )
 enhancementString      [0] IMPLICIT VisibleString,
ENDIF
IF ( dei )
 enhancementIndex       [1] IMPLICIT INTEGER,
ENDIF
 noEnhancement          NULL }
```

4.113 AcknowledgeEventNotification

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

AcknowledgeEventNotification-Response ::= NULL
```

4.114 GetAlarmSummary

```
GetAlarmSummary-Request ::= SEQUENCE {
 enrollmentsOnly        [0] IMPLICIT BOOLEAN DEFAULT TRUE,
 activeAlarmsOnly       [1] IMPLICIT BOOLEAN DEFAULT TRUE,
 acknowledgementFilter  [2] IMPLICIT INTEGER {
 not-acked                  (0), -- NOT-ACKED
 acked                      (1), -- ACKED
 all                        (2) -- ALL
 } DEFAULT not-acked,
 severityFilter         [3] IMPLICIT SEQUENCE {
 mostSevere                 [0] IMPLICIT Unsigned8,
 leastSevere                [1] IMPLICIT Unsigned8 } DEFAULT {
                                mostSevere 0, leastSevere 127 },
```

```
 continueAfter           [5] ObjectName OPTIONAL }

GetAlarmSummary-Response ::= SEQUENCE {
 listOfAlarmSummary  [0] IMPLICIT SEQUENCE OF AlarmSummary,
 moreFollows         [1] IMPLICIT BOOLEAN DEFAULT FALSE }

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
 both                             (3)  -- BOTH
 },
IF ( csr cspi )
 additionalDetail              [4] EN-Additional-Detail OPTIONAL,
                  -- shall not be transmitted if the value is NULL
ENDIF
 timeOfLastTransitionToActive [5] EventTime OPTIONAL,
 timeOfLastTransitionToIdle   [6] EventTime OPTIONAL }
```

4.115 GetAlarmEnrollmentSummary

```
GetAlarmEnrollmentSummary-Request ::= SEQUENCE {
 enrollmentsOnly         [0] IMPLICIT BOOLEAN DEFAULT TRUE,
 activeAlarmsOnly        [1] IMPLICIT BOOLEAN DEFAULT TRUE,
 acknowledgementFilter   [2] IMPLICIT INTEGER {
 not-acked                   (0), -- NOT-ACKED
 acked                       (1), -- ACKED
 all                         (2) -- ALL
 } DEFAULT not-acked,
 severityFilter          [3] IMPLICIT SEQUENCE {
 mostSevere                  [0] IMPLICIT Unsigned8,
 leastSevere                 [1] IMPLICIT Unsigned8 } DEFAULT
                                 { mostSevere 0, leastSevere 127 },
 continueAfter           [5] ObjectName OPTIONAL }

GetAlarmEnrollmentSummary-Response ::= SEQUENCE {
 listOfAlarmEnrollmentSummary [0] IMPLICIT SEQUENCE OF AlarmEnrollmentSummary,
 moreFollows                  [1] IMPLICIT BOOLEAN DEFAULT FALSE }

AlarmEnrollmentSummary ::= SEQUENCE {
 eventEnrollmentName          [0] ObjectName,
IF ( tpy )
 clientApplication            [2] ApplicationReference OPTIONAL,
ENDIF
 severity                     [3] IMPLICIT Unsigned8,
```

```
 currentState                  [4] IMPLICIT EC-State,
IF ( cspi )
 additionalDetail              [5] EN-Additional-Detail OPTIONAL,
                  -- shall not be transmitted if the value is NULL
ENDIF
 notificationLost              [6] IMPLICIT BOOLEAN DEFAULT FALSE,
 alarmAcknowledgmentRule       [7] IMPLICIT AlarmAckRule,
 enrollmentState               [8] IMPLICIT EE-State OPTIONAL,
 timeOfLastTransitionToActive  [9] EventTime OPTIONAL,
 timeActiveAcknowledged        [10] EventTime OPTIONAL,
 timeOfLastTransitionToIdle    [11] EventTime OPTIONAL,
 timeIdleAcknowledged          [12] EventTime OPTIONAL }
```

4.116 AttachToEventCondition

```
AttachToEventCondition ::= SEQUENCE {
 eventEnrollmentName       [0] ObjectName,
 eventConditionName        [1] ObjectName,
 causingTransitions        [2] IMPLICIT Transitions,
 acceptableDelay           [3] IMPLICIT Unsigned32 OPTIONAL }
```

4.117 EC-Class

```
EC-Class ::= INTEGER {
 network-triggered      (0), -- NETWORK-TRIGGERED,
 monitored              (1)  -- MONITORED
 }
```

4.118 EC-State

```
EC-State ::= INTEGER {
 disabled               (0), -- DISABLED
 idle                   (1), -- IDLE
 active                 (2)  -- ACTIVE
 }
```

4.119 EE-State

```
EE-State ::= INTEGER {
 disabled               (0), -- DISABLED
 idle                   (1), -- IDLE
 active                 (2), -- ACTIVE
 activeNoAckA           (3), -- ACTIVE-NO-ACK-A
 idleNoAckI             (4), -- IDLE-NO-ACK-I
 idleNoAckA             (5), -- IDLE-NO-ACK-A
 idleAcked              (6), -- IDLE-ACKED
 activeAcked            (7)  -- ACTIVE-ACKED
 }
```