

ECMA

标准信息通信系统

计算机支持电信应用协议 (CSTA) I

ECMA

标准信息 and 通信系统

计算机支持电信应用协议 (CSTA) I

历史简介

此标准定义了计算机支持电信应用协议，该协议是在OSI第7层计算机网络和电信网络之间的通信协议。此标准是ECMA-179计算机支持电信应用的附加标准。反映了ECMA成员关于CSTA标准第一阶段的协议。预计会有其他阶段。本标准是基于ECMA成员公司的实践经验，实用和广泛的共识。本标准从技术报告ECMA TR / 52计算机支持的电信应用而来。

**Usage Note regarding removing Year 2000 (Y2K) risks
in the CSTA Phase I and Phase II Protocol Standards**

1 January 1999

In both ITU-T Recs. X.208 (ASN.1 1988) and X.680 (ASN.1 1994), the definition of UTCTime, which is used in the CSTA Phase I and Phase II Protocol Standards (ECMA-180 and ECMA-218 respectively), is not Y2K-safe. The Year field (the YY field) is represented by a two-digit string, with no accompanying text mandating a Y2K-safe interpretation.

For the CSTA Phase I and Phase II Protocol Standards, the following interpretation of the Year field is mandated :

- If the YY component is in the range [00-49], the century is determined to be "20" (i.e. the year is 2000 to 2049);
- If the YY component is in the range [50-99], the century is determined to be "19" (i.e. the year is 1950 to 1999).

The risk has been removed in the CSTA Phase III Protocol Standard (ECMA-285) by replacing the references to "UTCTime" with references to "GeneralizedTime", which is Y2K-safe.

Table of contents

	Page
Section 1 - General	1
1 Scope	1
2 Conformance	1
2.1 Static requirements	1
2.2 Dynamic requirements	1
2.3 PICS requirement	1
3 References	2
4 Definitions	2
Section 2 - Protocol Structure for CSTA	3
5 CSTA service definition model	3
5.1 CSTA application layer structure	3
5.2 Remote operations	3
5.3 The CSTA service response	3
5.4 Cross referencing of event reports	4
5.5 Handling of private data	4
6 Interconnection service boundary	4
7 Security	4
Section 3 - CSTA Protocol	5
8 Association management	5
9 Switching function services	5
9.1 Alternate call	6
9.2 Answer call	7
9.3 Call completion	8
9.4 Clear call	9
9.5 Clear connection	10
9.6 Conference call	11
9.7 Consultation call	12
9.8 Divert call	13
9.9 Hold call	14
9.10 Make call	15
9.11 Make Predictive call	16
9.12 Query device	17
9.13 Reconnect call	18
9.14 Retrieve call	19
9.15 Set feature	20
9.16 Transfer call	21
10 Switching function events	22
10.1 Call events	23
10.1.1 Call cleared	23

10.1.2	Conferenced	24
10.1.3	Connection cleared	25
10.1.4	Delivered	26
10.1.5	Diverted	27
10.1.6	Established	28
10.1.7	Failed	29
10.1.8	Held	30
10.1.9	Network reached	31
10.1.10	Originated	32
10.1.11	Queued	33
10.1.12	Retrieved	34
10.1.13	Service initiated	35
10.1.14	Transferred	36
10.2	Feature events	37
10.2.1	Call information	37
10.2.2	Do not disturb	38
10.2.3	Forwarding	39
10.2.4	Message waiting	40
10.3	Agent state events	41
10.3.1	Logged on	41
10.3.2	Logged off	42
10.3.3	Not ready	43
10.3.4	Ready	44
10.3.5	Work not ready	45
10.3.6	Work ready	46
10.4	Maintenance events	47
10.4.1	Back in service	47
10.4.2	Out of service event	48
10.5	Private events	48
11	Computing function services	49
11.1	Route request	49
11.2	Re-route request	50
11.3	Route select request	51
11.4	Route used request	52
11.5	Route end request	53
12	Bidirectional services	54
12.1	Escape service	54
12.2	System status	55
13	Status reporting services	56
13.1	Monitor start	56
13.2	Change monitor filter	57
13.3	Monitor stop	58
13.4	Snapshot device	59

13.5	Snapshot call	60
14	Switching function errors	61
15	Switching event cause values	63
16	CSTA data types	64
16.1	Switching function objects	65
16.2	Device identifiers	66
16.3	Call and connection identifiers	68
16.4	Connection states	69
16.5	Status reporting	70
16.6	Device and feature types and other parameters	73
16.7	Security service	77
16.8	Common extensions	78
Annex A	Protocol Implementation Conformance Statement (PICS) Proforma	79

Section 1 - General

1 Scope

This Standard specifies application protocol data units (APDUs) for the services described in ECMA-179, *Services for Computer-Supported Telecommunications Applications*. The field of application of this Standard is the interconnection of switches and computers in a private telecommunications environment.

Section 2 (clause 5 to clause 7 inclusive) describes the concepts underlying the Remote Operations model, notation and service. Section 3 (clause 8 to clause 16 inclusive) contains CSTA-specific protocol details and forms the main body of this Standard.

The protocol defined in this Standard operates in the context of an application association. It is assumed that such an application association exists via mechanisms that are not defined in this Standard.

2 Conformance

A manufacturer may select any part (one or more operations) of the CSTA Protocol, as specified in this Standard, for implementation on a system.

A system is in conformance with one or more of the CSTA operations if those operations are implemented according to the definitions in this Standard.

A Protocol Implementation Conformance Statement (PICS) shall be used to specify the operations which are provided by a particular implementation. A PICS shall also specify the parameter options which are used.

A system claiming to implement one or more operations specified within this Standard shall comply with the relevant requirements in clauses 5 to 16 inclusive.

2.1 Static requirements

A system claiming conformance shall support the transfer syntax (derived from the encoding rules specified in CCITT Rec. X.209) named {joint-iso-ccitt(2) asn1(1) basic-encoding(1)}; for the purpose of generating and interpreting CSTA protocol information as defined by the abstract syntax "CSTA-ASN.1-Object-Descriptor" for the operations supported.

2.2 Dynamic requirements

A system claiming conformance shall:

- i) follow the procedures as specified in this Standard, and in ECMA-179, relevant to each CSTA operation that the system claims to implement, and
- ii) satisfy the definitions, as specified in ECMA-179, relevant to each CSTA service that the system claims to implement.

2.3 PICS requirement

The following shall be stated by the implementer when defining a PICS corresponding to an application or implementation:

- i) which CSTA operations, as defined in ECMA-179, are supported by the system for the particular implementation,
- ii) which optional parameters are supported by the PDUs belonging to the supported operations, and
- iii) the types and ranges of values for all the parameters supported.

A PICS Proforma is given in annex A.

3 References

ECMA-138	Security in Open Systems - Data Elements and Service Definitions (1989)
ECMA-155	Addressing in Private Telecommunication Networks (1991)
ECMA-179	Services for Computer-Supported Telecommunications Applications (CSTA) (1992)
ENV 41007	Definitions of Terms in Private Telecommunications Networks
ISO 9545	Open Systems Interconnection - Application Layer Structure
CCITT X.208 / ISO 8824	Specification of abstract syntax notation
CCITT X.209 / ISO 8825	Specification of basic encoding rules for the abstract syntax notation
CCITT X.217 / ISO 8649	Association control service definition
CCITT X.219 / ISO 9072-1	Remote operations - Part 1: model notation and service definition
CCITT X.227 / ISO 8650	Association control protocol specification
CCITT X.229 / ISO 9072-2	Remote operations - Part 2: protocol specification
CCITT E.164	Numbering plan for ISDN

All the CCITT Recommendations referenced above are the 1988 (Blue Book) versions.

4 Definitions

CSTA-specific terminology is defined in ECMA-179. For the purposes of this Standard, the following additional definitions, defined elsewhere, shall apply:

Remote Operations	as per CCITT Rec. X.219
Application Association	as per CCITT Rec. X.217
Application Context	as per CCITT Rec. X.217
Private Telecommunications Network	as per CENELEC ENV 41007

Section 2 - Protocol Structure for CSTA

5 CSTA service definition model

5.1 CSTA application layer structure

The CSTA application layer structure conforms to the model as described in ISO 9545.

5.2 Remote operations

The services of CSTA are modelled as Remote Operations as described in CCITT Rec. X.219. Typically, one entity requests that a particular operation be performed; the other entity attempts to perform the operation and responds to the requestor. Consequently the operation of the protocol is an elementary request/reply interaction, supported within the OSI application layer, and carried out within the context of an application-association.

For some of the CSTA services, the entity to which the request is sent generates a reply which can indicate success or failure. For these services, CSTA shall adopt the Operations Class 2, defined in Rec. X.219 as:

- Asynchronous, reporting success or failure (result or error).

For some of the CSTA services, the entity to which the request is sent generates a reply which can only indicate failure. For these services, CSTA shall adopt the Operations Class 3, defined in Rec. X.219 as:

- Asynchronous, reporting failure (error) only, if any.

For some of the CSTA services, particularly the ongoing reporting of events, no reply is generated. For these services, CSTA shall adopt the Operations Class 5, defined in Rec. X.219 as:

- Asynchronous, outcome not reported.

The protocol description for the particular service defines the relevant class of the Operation used for that service.

CSTA shall correlate the single response, denoting success or failure, with the originating request by using the mechanisms within the ROSE protocol.

5.3 The CSTA service response

CSTA employs a generic response mechanism which is, in principle, decoupled from the specifics of the switching activity. The following points describe the operation of the CSTA service response:

1. Specific services may have an unconfirmed mode where responses to correct requests are not returned.
2. The server shall check the correctness of the request (e.g. syntactical checks) before issuing the response. Incorrect requests shall result in an error response, even in the unconfirmed mode.
3. If a response is sent before the action requested by the service is completed (i.e. the response is a service request acknowledgement) event reporting may be used to keep track of the subsequent server activity.
4. The precise moment at which the response is generated in relation to the switching activity is implementation and service dependent.
 - i) Some implementations may generate the response after checking the correctness of the request and at the point they initiate the request.
 - ii) Other implementations may delay the response until the service has completed (or is guaranteed to complete). In this case, a failure of the switching request will be reflected in the response.

NOTE 1

Irrespective of implementation details, when an operation succeeds the same event reports are generated if the necessary monitoring has been established. As an example, a Held-Event (if selected) is always reported in addition to the response to a successful request (even in those implementations that delay the

response until the Hold operation was complete). In a given context, and with appropriate monitoring in place, an operation generates the same set of event reports, whether it was invoked manually or with CSTA service requests from the computer.

5.4 Cross referencing of event reports

A computer application process may need to cross reference a CSTAEventReport to one of the following:

- a) a CSTA Object ID (Call ID or Device ID),
- b) an earlier Monitor request, or
- c) one of many Monitor requests (pertaining to the same CSTA Object).

For the above scenarios the necessary cross referencing function may be fulfilled by use of the parameter "MonitorCrossRefID". The content of MonitorCrossRefID depends upon the context and it may be one of the following: Call ID, StaticDevice ID, or another independently switch-managed static identifier. The independent identifier may have a unique correlation to either: one device, one call, or one monitor request.

The switching system limit on the number of monitor requests on one CSTA Object (Call or Device) is an implementation option. This Standard does not stipulate how many monitor requests on one object are to be supported by the switch. If using Static Device or Call identifiers the limit can only be one.

5.5 Handling of private data

If an entity receives the parameter CSTAPrivateData, and it can not recognize the information contained, the parameter shall be discarded, and the rest of the message shall be processed.

6 内部服务连接边界

The protocol in this Standard is an OSI application layer protocol and uses the Remote Operations protocol defined in CCITT Rec. X.229. The Remote Operations protocol assumes certain services are provided by the underlying layers, and these services are also assumed by the protocol for CSTA.

7 安全

此协议提供了CSTA安全传输结构 PDUs。此参数是由ECMA-138安全PDU构成。

Section 3 - CSTA 协议

8 Association management

The protocol in this Standard operates in the context of an application association. It is assumed that such an application association exists via mechanisms not defined in this Standard.

9 交换功能服务

此章节定义了CSTA交换服务协议，使用了ASN.1文本描述ECMA-179提供的CSTA服务。协议的编码定义在CCITT Rec. X.209 “使用抽象语法标记指定了基础编码规则”

注意 2

The range of services supported on a particular association is specified in the application context at association time.

9.1 切换呼叫

```
CSTA-alternate-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) alternate-call( 1) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionDetails FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

alternateCall      OPERATION
    ARGUMENT AlternateCallArgument
    RESULT      AlternateCallResult
    ERRORS      {universalFailure}
::= 1

AlternateCallArgument ::=
    CHOICE
    {callsInvolved      ConnectionDetails,
     SEQUENCE
     {callsInvolved      ConnectionDetails,
      extensions          CSTACommonArguments          OPTIONAL} }

AlternateCallResult ::=
    CHOICE
    {extensions          CSTAPrivateData,
     noData              NULL}

END -- of CSTA-alternate-call
```

9.2 Answer call

```
CSTA-answer-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) answer-call( 2) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

answerCall OPERATION
    ARGUMENT AnswerCallArgument
    RESULT    AnswerCallResult
    ERRORS    {universalFailure}
::= 2

AnswerCallArgument ::=
    CHOICE
    {callToBeAnswered    ConnectionID,
     SEQUENCE
     {callToBeAnswered ConnectionID,
      extensions        CSTACommonArguments    OPTIONAL} }

AnswerCallResult ::=
    CHOICE
    {extensions        CSTAPrivateData,
     noData            NULL}

END -- of CSTA-answer-call
```

9.3 Call completion

```
CSTA-call-completion
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-completion( 3) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
FeatureInfo FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

callCompletion OPERATION
    ARGUMENT CallCompletionArgument
    RESULT    CallCompletionResult
    ERRORS    {universalFailure}
::= 3

CallCompletionArgument ::=
    CHOICE
    {featureInfo          FeatureInfo,
     SEQUENCE
     {featureInfo        FeatureInfo,
      extensions         CSTACommonArguments          OPTIONAL} }

CallCompletionResult ::=
    CHOICE
    {extensions          CSTAPrivateData,
     noData              NULL}

END -- of CSTA-call-completion
```

9.4 Clear call

```
CSTA-clear-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) clear-call( 4) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

clearCall OPERATION
    ARGUMENT ClearCallArgument
    RESULT    ClearCallResult
    ERRORS    {universalFailure}
::= 4

ClearCallArgument ::=
    CHOICE
    {callToBeCleared      ConnectionID,
     SEQUENCE
     {callToBeCleared     ConnectionID,
      extensions          CSTACommonArguments      OPTIONAL} }

ClearCallResult ::=
    CHOICE
    {extensions          CSTAPrivateData,
     noData              NULL}

END -- of CSTA-clear-call
```


9.6 Conference call

```
CSTA-conference-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) conference-call( 6) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
ConnectionDetails FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
ConnectionList FROM CSTA-connection-states
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) connection-states( 125) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

conferenceCall OPERATION
    ARGUMENT ConferenceCallArgument
    RESULT    ConferenceCallResult
    ERRORS    {universalFailure}
::= 6

ConferenceCallArgument ::=
    CHOICE
    {callsInvolved      ConnectionDetails,
     SEQUENCE
     {callsInvolved      ConnectionDetails,
      extensions          CSTACommonArguments          OPTIONAL} }

ConferenceCallResult ::=
    SEQUENCE
    {conferenceCall      ConnectionID,
     connections          ConnectionList          OPTIONAL,
     extensions           CSTAPrivateData          OPTIONAL}

END -- of CSTA-conference-call
```

9.7 Consultation call

```
CSTA-consultation-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) consultation-call( 7) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CalledDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

consultationCall OPERATION
    ARGUMENT ConsultationCallArgument
    RESULT    ConsultationCallResult
    ERRORS    {universalFailure}
::= 7

ConsultationCallArgument ::=
    SEQUENCE
    {existingCall
     calledDirectoryNumber
     extensions
     ConnectionID,
     CalledDeviceID,
     CSTACommonArguments OPTIONAL}

ConsultationCallResult ::=
    CHOICE
    {initiatedCall
     SEQUENCE
     {initiatedCall
      extensions
      ConnectionID,
      CSTAPrivateData
      OPTIONAL} }

END -- of CSTA-consultation-call
```


9.8 Divert call

```
CSTA-divert-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) divert-call( 8) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
DivertInfo FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

divertCall OPERATION
    ARGUMENT DivertCallArgument
    RESULT    DivertCallResult
    ERRORS    {universalFailure}
::= 8

DivertCallArgument ::=
    CHOICE
    {divertInfo          DivertInfo,
     SEQUENCE
     {divertInfo
      extensions          DivertInfo,
                          CSTACommonArguments          OPTIONAL} }

DivertCallResult ::=
    CHOICE
    {extensions          CSTAPrivateData,
     noData              NULL}

END -- of CSTA-divert-call
```

9.9 Hold call

```
CSTA-hold-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) hold-call( 9) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
ReserveConnection FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

holdCall OPERATION
    ARGUMENT HoldCallArgument
    RESULT    HoldCallResult
    ERRORS    {universalFailure}
::= 9

HoldCallArgument ::=
    SEQUENCE
    {callToBeHeld
      connectionReservation
      extensions
      ConnectionID,
      ReserveConnection
      CSTACommonArguments
      DEFAULT FALSE,
      OPTIONAL}

HoldCallResult ::=
    CHOICE
    {extensions
      noData
      CSTAPrivateData,
      NULL}

END -- of CSTA-hold-call
```

9.10 Make call

```
CSTA-make-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) make-call( 10) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
DeviceID, CalledDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

makeCall OPERATION
    ARGUMENT MakeCallArgument
    RESULT    MakeCallResult
    ERRORS    {universalFailure}
::= 10

MakeCallArgument ::=
    SEQUENCE
    {callingDevice      DeviceID,
     calledDirectoryNumber CalledDeviceID,
     extensions         CSTACommonArguments OPTIONAL}

MakeCallResult ::=
    CHOICE
    {initiatedCall      ConnectionID,
     SEQUENCE
     {initiatedCall     ConnectionID,
      extensions        CSTAPrivateData OPTIONAL}}

END -- of CSTA-make-call
```

9.11 Make Predictive call

```
CSTA-make-predictive-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) make-predictive-call( 11) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
AllocationState FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
DeviceID, CalledDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

makePredictiveCall OPERATION
    ARGUMENT MakePredictiveCallArgument
    RESULT    MakePredictiveCallResult
    ERRORS    {universalFailure}
::= 11

MakePredictiveCallArgument ::=
    SEQUENCE
    {callingDevice          DeviceID,
     calledDirectoryNumber CalledDeviceID,
     allocation             AllocationState          DEFAULT callDelivered,
     extensions             CSTACommonArguments     OPTIONAL}

MakePredictiveCallResult ::=
    CHOICE
    {initiatedCall          ConnectionID,
     SEQUENCE
     {initiatedCall         ConnectionID,
      extensions            CSTAPrivateData          OPTIONAL} }

END -- of CSTA-make-predictive-call
```

9.12 Query device

```
CSTA-query-device
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) query-device( 12) }

DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }

-- Data Types --
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
QueryDeviceFeature, QueryDeviceInformation FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

queryDevice OPERATION
    ARGUMENT QueryDeviceArgument
    RESULT    QueryDeviceResult
    ERRORS    {universalFailure}

 ::= 12

QueryDeviceArgument ::=
    SEQUENCE
    {device          DeviceID,
     feature         QueryDeviceFeature,
     extensions      CSTACommonArguments OPTIONAL}

QueryDeviceResult ::=
    CHOICE
    {deviceInformation QueryDeviceInformation,
     SEQUENCE
     {deviceInformation QueryDeviceInformation,
      extensions        CSTAPrivateData OPTIONAL} }

END -- of CSTA-query-device
```

9.13 Reconnect call

```
CSTA-reconnect-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) reconnect-call( 13) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionDetails FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACCommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

reconnectCall OPERATION
    ARGUMENT ReconnectCallArgument
    RESULT    ReconnectCallResult
    ERRORS    {universalFailure}
::= 13

ReconnectCallArgument ::=
    CHOICE
    {reconnectInfo      ConnectionDetails,
     SEQUENCE
     {reconnectInfo     ConnectionDetails,
      extensions        CSTACCommonArguments      OPTIONAL} }

ReconnectCallResult ::=
    CHOICE
    {extensions        CSTAPrivateData,
     noData            NULL}

END -- of CSTA-reconnect-call
```


9.15 Set feature

```
CSTA-set-feature
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) set-feature( 15) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
SetDeviceFeature FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

setFeature OPERATION
    ARGUMENT SetFeatureArgument
    RESULT    SetFeatureResult
    ERRORS    {universalFailure}
::= 15

SetFeatureArgument ::=
    SEQUENCE
    {device          DeviceID,
     feature         SetDeviceFeature,
     extensions      CSTACommonArguments OPTIONAL}

SetFeatureResult ::=
    CHOICE
    {extensions      CSTAPrivateData,
     noData          NULL}

END -- of CSTA-set-feature
```


9.16 Transfer call

```
CSTA-transfer-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) transfer-call( 16) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
ConnectionDetails FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
ConnectionList FROM CSTA-connection-states
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) connection-states( 125) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

transferCall OPERATION
    ARGUMENT TransferCallArgument
    RESULT    TransferCallResult
    ERRORS    {universalFailure}
::= 16

TransferCallArgument ::=
    CHOICE
    {transferInfo          ConnectionDetails,
     SEQUENCE
     {transferInfo          ConnectionDetails,
      extensions            CSTACommonArguments OPTIONAL} }

TransferCallResult ::=
    SEQUENCE
    {transferredCall        ConnectionID                OPTIONAL,
     connections            ConnectionList              OPTIONAL,
     extensions             CSTAPrivateData             OPTIONAL}

END -- of CSTA-transfer-call
```

10 交换功能事件

这一章定义了CSTA事件服务协议。使用ASN.1. 语言描述了在ECMA-179中定义的CSTA服务

注意 3

The range of services supported on a particular association is specified in the application context at association time.

The structure of an event is as described by the remote operation 'cSTAEventReport'. This is a ROSE class 5 operation, which has an argument containing the event type and other information associated with that event. The EVENT macro is used to define events in this standard. This macro provides a shorthand method to define the data types associated with a particular event. Each use of the macro defines (after the EVENTINFO keyword) the data type that replaces the keyword 'ANY' in the cSTAEventReport operation for that particular event value.

```
CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions(21) }
DEFINITIONS ::=
BEGIN
EXPORTS
EVENT;
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
MonitorCrossRefID FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) };

cSTAEventReport OPERATION
ARGUMENT      CSTAEventReportArgument
::= 21

CSTAEventReportArgument ::=
    SEQUENCE
    {crossRefIdentifier      MonitorCrossRefID, -- allocated by switch
     eventType              EventTypeID,
     eventInfo              ANY DEFINED BY eventType,
     extensions             CSTAPrivateData          OPTIONAL}

EventTypeID ::= CHOICE { cSTAform      [0] IMPLICIT value(EVENT) }

EVENT MACRO ::=
BEGIN
    TYPE NOTATION      ::= "EVENTINFO" type | empty
    VALUE NOTATION     ::= value(VALUE INTEGER)
END -- of EVENT macro

END -- of CSTA-event-report-definitions
```

10.1 Call events

Each event contains a ConnectionID which identifies the object of interest. Other parameters may also be used to identify the relevant devices.

Events may result from a call interacting with switch features that had been previously set or invoked. These features and their settings may be reflected in an Event as a value of the EventCause parameter.

10.1.1 Call cleared

```
CSTA-call-cleared-event
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-cleared-event( 22) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) connection-states( 125) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-causes( 121) };

callCleared      EVENT
EVENTINFO        CallClearedEventInfo
::= 1

CallClearedEventInfo ::=
SEQUENCE
{clearedCall      ConnectionID,
 localConnectionInfo LocalConnectionState
 cause            EventCause
                  OPTIONAL,
                  OPTIONAL}
```

END -- of CSTA-call-cleared-event

10.1.2 Conferenced

```
CSTA-conferenced-event
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) conferenced-event( 23) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState, ConnectionList FROM CSTA-connection-states
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID FROM CSTA-device-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-causes( 121) };

conferenced      EVENT
EVENTINFO        ConferencedEventInfo
::= 2

ConferencedEventInfo ::=
  SEQUENCE
  {primaryOldCall      ConnectionID,
   secondaryOldCall    ConnectionID          OPTIONAL,
   confController      SubjectDeviceID,
   addedParty          SubjectDeviceID,
   conferenceConnections ConnectionList      OPTIONAL,
   localConnectionInfo LocalConnectionState OPTIONAL,
   cause               EventCause            OPTIONAL}

END -- of CSTA-conferenced-event
```

10.1.3 Connection cleared

```
CSTA-connection-cleared-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-cleared-event( 24) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

connectionCleared EVENT
EVENTINFO      ConnectionClearedEventInfo
::= 3

ConnectionClearedEventInfo ::=
SEQUENCE
{droppedConnection      ConnectionID,
 releasingDevice        SubjectDeviceID,
 localConnectionInfo    LocalConnectionState      OPTIONAL,
 cause                  EventCause                OPTIONAL}

END -- of CSTA-connection-cleared-event
```

10.1.4 Delivered

```
CSTA-delivered-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) delivered-event( 25) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CallingDeviceID, CalledDeviceID, RedirectionDeviceID
FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

delivered          EVENT
EVENTINFO          DeliveredEventInfo
::= 4

DeliveredEventInfo ::=
SEQUENCE
{connection          ConnectionID,
 alertingDevice      SubjectDeviceID,
 callingDevice        CallingDeviceID,
 calledDevice         CalledDeviceID,
 lastRedirectionDevice RedirectionDeviceID,
 localConnectionInfo LocalConnectionState OPTIONAL,
 cause               EventCause          OPTIONAL}

END -- of CSTA-delivered-event
```

10.1.5 Diverted

```
CSTA-diverted-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) diverted-event( 26) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CalledDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

diverted          EVENT
EVENTINFO         DivertedEventInfo
::= 5

DivertedEventInfo ::=
SEQUENCE
{connection          ConnectionID          OPTIONAL,
 divertingDevice     SubjectDeviceID,
 newDestination      CalledDeviceID,
 localConnectionInfo LocalConnectionState  OPTIONAL,
 cause              EventCause            OPTIONAL}

END -- of diverted-event
```

10.1.6 Established

```
CSTA--established-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) established-event( 27) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CalledDeviceID, CallingDeviceID, RedirectionDeviceID
FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

established      EVENT
EVENTINFO      EstablishedEventInfo
::= 6

EstablishedEventInfo ::=
SEQUENCE
{establishedConnection      ConnectionID,
 answeringDevice            SubjectDeviceID,
 callingDevice              CallingDeviceID,
 calledDevice              CalledDeviceID,
 lastRedirectionDevice      RedirectionDeviceID,
 localConnectionInfo        LocalConnectionState      OPTIONAL,
 cause                     EventCause                 OPTIONAL}

END -- of CSTA-established-event
```


10.1.7 Failed

```
CSTA-failed-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) failed-event( 28) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CalledDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

failed          EVENT
EVENTINFO       FailedEventInfo
::= 7

FailedEventInfo ::=
SEQUENCE
{failedConnection          ConnectionID,
 failingDevice             SubjectDeviceID,
 calledDevice              CalledDeviceID,
 localConnectionInfo      LocalConnectionState
 cause                    EventCause
                           OPTIONAL,
                           OPTIONAL}
```

END -- of CSTA-failed-event

10.1.8 Held

```
CSTA-held-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) held-event( 29) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

held          EVENT
EVENTINFO     HeldEventInfo
::= 8

HeldEventInfo ::=
SEQUENCE
{heldConnection          ConnectionID,
 holdingDevice           SubjectDeviceID,
 localConnectionInfo     LocalConnectionState OPTIONAL,
 cause                   EventCause          OPTIONAL}

END -- of CSTA-held-event
```

10.1.9 Network reached

```
CSTA-network-reached-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) network-reached-event( 30) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CalledDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

networkReached EVENT
EVENTINFO      NetworkReachedEventInfo
::= 9

NetworkReachedEventInfo ::=
SEQUENCE
{connection          ConnectionID,
 trunkUsed           SubjectDeviceID,
 calledDevice        CalledDeviceID,
 localConnectionInfo LocalConnectionState OPTIONAL,
 cause              EventCause          OPTIONAL}
```

END -- of CSTA-network-reached-event

10.1.10 Originated

```
CSTA-originated-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) originated-event( 31) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID, CalledDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

originated      EVENT
EVENTINFO      OriginatedEventInfo
::= 10

OriginatedEventInfo ::=
SEQUENCE
{originatedConnection      ConnectionID,
 callingDevice              SubjectDeviceID,
 calledDevice               CalledDeviceID,
 localConnectionInfo        LocalConnectionState      OPTIONAL,
 cause                      EventCause                 OPTIONAL}

END -- of CSTA-originated-event
```

10.1.11 Queued

```

CSTA-queued-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) queued-event( 32) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) connection-states( 125) }
CalledDeviceID, CallingDeviceID, RedirectionDeviceID, SubjectDeviceID
  FROM CSTA-device-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
NoOfCallsInQueue FROM CSTA-device-feature-types
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
EventCause FROM CSTA-event-causes
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-causes( 121) };

queued          EVENT
EVENTINFO       QueuedEventInfo
::= 11

QueuedEventInfo ::=
SEQUENCE
{queuedConnection      ConnectionID,
 queue                 SubjectDeviceID,
 callingDevice          CallingDeviceID,
 calledDevice           CalledDeviceID,
 lastRedirectionDevice  RedirectionDeviceID,
 numberedQueued         NoOfCallsInQueue
 localConnectionInfo   LocalConnectionState
 cause                 EventCause
                                OPTIONAL,
                                OPTIONAL,
                                OPTIONAL}

END -- of CSTA-queued-event

```

10.1.12 Retrieved

```
CSTA-retrieved-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) retrieved-event( 33) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

retrieved          EVENT
EVENTINFO          RetrievedEventInfo
::= 12

RetrievedEventInfo ::=
SEQUENCE
{retrievedConnection      ConnectionID,
 retrievingDevice          SubjectDeviceID,
 localConnectionInfo       LocalConnectionState      OPTIONAL,
 cause                     EventCause                 OPTIONAL}

END -- of CSTA-retrieved-event
```

10.1.13 Service initiated

```
CSTA-service-initiated-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) service-initiated-event( 34) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

serviceInitiated    EVENT
EVENTINFO           ServiceInitiatedEventInfo
::= 13

ServiceInitiatedEventInfo ::=
SEQUENCE
{initiatedConnection          ConnectionID,
 localConnectionInfo          LocalConnectionState
 cause                        EventCause
                             OPTIONAL,
                             OPTIONAL}
```

10.1.14 Transferred

```
CSTA-transferred-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) transferred-event( 35) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
LocalConnectionState, ConnectionList FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
EventCause FROM CSTA-event-causes
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-causes( 121) };

transferred          EVENT
EVENTINFO            TransferredEventInfo
::= 14

TransferredEventInfo ::=
SEQUENCE
{primaryOldCall          ConnectionID,
 secondaryOldCall        ConnectionID          OPTIONAL,
 transferringDevice       SubjectDeviceID,
 transferredDevice        SubjectDeviceID,
 transferredConnections   ConnectionList        OPTIONAL,
 localConnectionInfo     LocalConnectionState  OPTIONAL,
 cause                   EventCause            OPTIONAL}

END -- of CSTA-transferred-event
```


10.2 Feature events

10.2.1 Call information

```
CSTA-call-information-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-information-event( 41) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AccountInfo, AuthCode FROM CSTA-device-feature-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) };

callInformation      EVENT
EVENTINFO            CallInformationEventInfo
::= 101

CallInformationEventInfo ::=
SEQUENCE
{connection          ConnectionID,
 device              SubjectDeviceID,
 accountInfo         [10] IMPLICIT AccountInfo      OPTIONAL,
 authorisationCode   [11] IMPLICIT AuthCode          OPTIONAL}

END -- of CSTA-call-information-event
```

10.2.2 Do not disturb

```
CSTA-do-not-disturb-event
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) do-not-disturb-event( 42) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

doNotDisturb      EVENT
EVENTINFO         DoNotDisturbEventInfo
::= 102

DoNotDisturbEventInfo ::=
  SEQUENCE
    {device                               SubjectDeviceID,
     doNotDisturbOn                       BOOLEAN}
```

END -- of CSTA-do-not-disturb-event

10.2.3 Forwarding

```
CSTA-forwarding-event
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) forwarding-event( 43) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
ForwardParameter FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
SubjectDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

forwarding      EVENT
EVENTINFO      ForwardingEventInfo
::= 103

ForwardingEventInfo ::=
    SEQUENCE
    {device
      forwardingInformation
      SubjectDeviceID,
      ForwardParameter}

END -- of CSTA-forwarding-event
```

10.2.4 Message waiting

```
CSTA-message-waiting-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) message-waiting-event( 44) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
CalledDeviceID, SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

messageWaiting  EVENT
EVENTINFO      MessageWaitingEventInfo
::= 104

MessageWaitingEventInfo ::=
  SEQUENCE
    {deviceForMessage      CalledDeviceID,
     invokingDevice        SubjectDeviceID,
     messageWaitingOn      BOOLEAN}

END -- of CSTA-message-waiting-event
```

10.3 Agent state events

10.3.1 Logged on

```
CSTA-logged-on-event
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) logged-on-event( 51) }

DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AgentID, AgentGroup, AgentPassword FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

loggedOn          EVENT
EVENTINFO         LoggedOnEventInfo
::= 201

LoggedOnEventInfo ::=
SEQUENCE
{agentDevice          SubjectDeviceID,
 agentID              [10] IMPLICIT AgentID          OPTIONAL,
 agentGroup           AgentGroup                      OPTIONAL,
 password             [11] IMPLICIT AgentPassword     OPTIONAL}

END -- of CSTA-logged-on-event
```

10.3.2 Logged off

```
CSTA-logged-off-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) logged-off-event( 52) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AgentID, AgentGroup FROM CSTA-device-feature-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

loggedOff          EVENT
EVENTINFO          LoggedOffEventInfo
::= 202

LoggedOffEventInfo ::=
  SEQUENCE
    {agentDevice          SubjectDeviceID,
     agentID              [10] IMPLICIT AgentID OPTIONAL,
     agentGroup           AgentGroup          OPTIONAL}

END -- of CSTA-logged-off-event
```

10.3.3 Not ready

```
CSTA-not-ready-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) not-ready-event( 53) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123)}
AgentID FROM CSTA-device-feature-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

notReady          EVENT
EVENTINFO         NotReadyEventInfo
::= 203

NotReadyEventInfo ::=
  SEQUENCE
    {agentDevice          SubjectDeviceID,
     agentID              [10] IMPLICIT AgentID          OPTIONAL}

END -- of CSTA-not-ready-event
```

10.3.4 Ready

```
CSTA-ready-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) ready-event( 54) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AgentID FROM CSTA-device-feature-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

ready          EVENT
EVENTINFO      ReadyEventInfo
::= 204

ReadyEventInfo ::=
  SEQUENCE
    {agentDevice          SubjectDeviceID,
     agentID              [10] IMPLICIT AgentID          OPTIONAL}

END -- of CSTA-ready-event
```


10.3.5 Work not ready

```
CSTA-work-not-ready-event
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) work-not-ready-event( 55) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AgentID FROM CSTA-device-feature-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

workNotReady      EVENT
EVENTINFO          WorkNotReadyEventInfo
::= 205

WorkNotReadyEventInfo ::=
  SEQUENCE
    {agentDevice          SubjectDeviceID,
     agentID              [10] IMPLICIT AgentID          OPTIONAL}

END -- of CSTA-work-not-ready-event
```

10.3.6 Work ready

```
CSTA-work-ready-event
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) work-ready-event( 56) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
SubjectDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
AgentID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) };

workReady          EVENT
EVENTINFO          WorkReadyEventInfo
::= 206

WorkReadyEventInfo ::=
    SEQUENCE
    {agentDevice          SubjectDeviceID,
     agentID              [10] IMPLICIT AgentID          OPTIONAL}

END -- of CSTA-work-ready-event
```

10.4 Maintenance events

10.4.1 Back in service

```
CSTA-back-in-service-event
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) back-in-service-event( 61) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
EventCause FROM CSTA-event-causes
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-causes( 121) };

backInService      EVENT
EVENTINFO          BackInServiceEventInfo
::= 301

BackInServiceEventInfo ::=
    SEQUENCE
    {device          DeviceID,
     cause           EventCause      OPTIONAL}

END -- of CSTA-back-in-service-event
```

10.4.2 Out of service event

```
CSTA-out-of-service-event
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) out-of-service-event( 62) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-report-definitions( 21) }
-- Data Types --
DeviceID FROM CSTA-device-identifiers
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
EventCause FROM CSTA-event-causes
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-causes( 121) };

outOfService      EVENT
EVENTINFO         OutOfServiceEventInfo
::= 302

OutOfServiceEventInfo ::=
  SEQUENCE
  { device          DeviceID,
    cause           EventCause      OPTIONAL}

END -- of CSTA-out-of-service-event
```

10.5 Private events

```
CSTA-private-event
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) private-event( 71) }
DEFINITIONS ::=
BEGIN
IMPORTS
EVENT FROM CSTA-event-report-definitions
  { iso( 1) identified-organization( 3) icd-ecma( 0012)
    standard( 0) csta( 180) version1( 1) event-report-definitions( 21) };

private          EVENT
EVENTINFO        PrivateEventInfo
::= 401

PrivateEventInfo ::= NULL
-- The actual encoding of the private event is added here, replacing NULL with another
-- valid ASN.1 type.

END -- of CSTA-private-event
```

11 Computing function services

This clause defines the protocol for the Computing function services of CSTA, using ASN.1. Text descriptions of the CSTA services are provided in ECMA-179.

NOTE 4

The range of services supported on a particular association is specified in the application context at association time.

11.1 Route request

```
CSTA-route-request
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) route-request( 81) }

DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CalledDeviceID, CallingDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
SelectValue, PriorityValue, SetUpValues, RoutingCrossRefID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACommonArguments FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

routeRequest OPERATION
    ARGUMENT RouteRequestArgument
    ERRORS    {universalFailure}
::= 31

RouteRequestArgument ::=
    SEQUENCE
        {crossRefIdentifier          RoutingCrossRefID,
         currentRoute                CalledDeviceID,
         callingDevice               CallingDeviceID    OPTIONAL,
         routedCall                  ConnectionID       OPTIONAL,
         routeSelAlgorithm           SelectValue        OPTIONAL,
         priority                    PriorityValue       OPTIONAL,
         setupInformation             SetUpValues        OPTIONAL,
         extensions                   CSTACommonArguments OPTIONAL}

END -- of CSTA-route-request
```

11.2 Re-route request

```
CSTA-re-route-request
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) re-route-request( 82) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
RoutingCrossRefID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACommonArguments FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

reRouteRequest OPERATION
    ARGUMENT ReRouteRequestArgument
    ERRORS    {universalFailure}
::= 32

ReRouteRequestArgument ::=
    SEQUENCE
        {crossRefIdentifier    RoutingCrossRefID,
         extensions            CSTACommonArguments          OPTIONAL}

END -- of CSTA-re-route-request
```

11.3 Route select request

```
CSTA-Route-select-request
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) route-select-request( 83) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CalledDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
RouteUsedFlag, RetryValue, SetUpValues, RoutingCrossRefID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACCommonArguments FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

routeSelectRequest      OPERATION
    ARGUMENT             RouteSelectRequestArgument
    ERRORS                {universalFailure}
::= 33

RouteSelectRequestArgument ::=
    SEQUENCE
    {crossRefIdentifier      RoutingCrossRefID,
     routeSelected          CalledDeviceID,
     remainRetry            RetryValue      OPTIONAL,
     setupInformation       SetUpValues     OPTIONAL,
     routeUsedReq          RouteUsedFlag   OPTIONAL,
     extensions             CSTACCommonArguments OPTIONAL}

END -- of CSTA-route-select-request
```

11.4 Route used request

```
CSTA-route-used-request
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) route-used-request( 84) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CallingDeviceID, CalledDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
DomainValue, RoutingCrossRefID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACCommonArguments FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

routeUsedRequest  OPERATION
    ARGUMENT RouteUsedRequestArgument
    ERRORS    {universalFailure}
::= 34

RouteUsedRequestArgument ::=
    SEQUENCE
        {crossRefIdentifier          RoutingCrossRefID,
         routeUsed                  CalledDeviceID,
         callingDevice              CallingDeviceID      OPTIONAL,
         domain                    DomainValue          OPTIONAL,
         extensions                 CSTACCommonArguments OPTIONAL}
```

END -- of CSTA-route-used-request

11.5 Route end request

```
CSTA-route-end-request
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) route-end-request( 85) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
RoutingCrossRefID FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
CSTACCommonArguments FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure, UniversalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

routeEndRequest    OPERATION
    ARGUMENT RouteEndRequestArgument
    ERRORS    {universalFailure}
::= 35

RouteEndRequestArgument ::=
    SEQUENCE
        {crossRefIdentifier          RoutingCrossRefID,
         errorValue                  UniversalFailure OPTIONAL,
         extensions                  CSTACCommonArguments OPTIONAL}

END -- of CSTA-route-end-request
```

12 Bidirectional services

This clause defines the protocol for the Bidirectional services of CSTA, using ASN.1. Text descriptions of the CSTA services are provided in ECMA-179.

NOTE 5

The range of services supported on a particular association is specified in the application context at association time.

12.1 Escape service

```
CSTA-escape-service
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) CSTA-escape-service( 91) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1(1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

escapeService      OPERATION
    ARGUMENT EscapeServiceArgument
    RESULT      EscapeServiceResult
    ERRORS      {universalFailure}
::= 51

EscapeServiceArgument ::=
    extensions          CSTACommonArguments

EscapeServiceResult ::=
    CHOICE
    {extensions          CSTAPrivateData,
     noData              NULL}

END -- of CSTA-escape-service
```

12.2 System status

```
CSTA-system-status
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) system-status( 92) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
SystemStatus FROM CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

systemStatus      OPERATION
    ARGUMENT SystemStatusArgument
    RESULT      SystemStatusResult
    ERRORS      {universalFailure}
::= 52

SystemStatusArgument ::=
    CHOICE
    {systemStatus      SystemStatus,
     SEQUENCE
     {systemStatus      SystemStatus,
      extensions        CSTACommonArguments      OPTIONAL} }

SystemStatusResult ::=
    CHOICE
    {extensions        CSTAPrivateData,
     noData            NULL}

END -- of CSTA-system-status
```

13 Status reporting services

This clause defines the protocol for the Status reporting services of CSTA, using ASN.1. Text descriptions of the CSTA services are provided in ECMA-179.

NOTE 6

The range of services supported on a particular association is specified in the application context at association time.

13.1 Monitor start

```

CSTA-monitor-start
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) monitor-start( 101) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
MonitorObject, MonitorFilter, MonitorType, MonitorCrossRefID FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) }
CSTACCommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

monitorStart      OPERATION
    ARGUMENT MonitorStartArgument
    RESULT      MonitorStartResult
    ERRORS      {universalFailure}
::= 71

MonitorStartArgument ::=
    SEQUENCE
    {monitorObject      MonitorObject,
     monitorFilter      MonitorFilter
                        OPTIONAL,
     monitorType        MonitorType
                        OPTIONAL,
     extensions          CSTACCommonArguments
                        OPTIONAL}

MonitorStartResult ::=
    SEQUENCE
    {crossRefIdentifier  MonitorCrossRefID,
     monitorFilter        MonitorFilter
                        OPTIONAL,
     extensions          CSTAPrivateData
                        OPTIONAL}

END -- of CSTA-monitor-start

```

13.2 Change monitor filter

```
CSTA-change-monitor-filter
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1 ( 1) change-monitor-filter( 102) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
MonitorFilter, MonitorCrossRefID FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) }
CSTACCommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

changeMonitorFilter      OPERATION
    ARGUMENT             ChangeMonitorFilterArgument
    RESULT               ChangeMonitorFilterResult
    ERRORS               {universalFailure}
::= 72

ChangeMonitorFilterArgument ::=
    SEQUENCE
    {monitorCrossRefID   MonitorCrossRefID,
     filterlist         MonitorFilter,
     extensions         CSTACCommonArguments OPTIONAL}

ChangeMonitorFilterResult ::=
    CHOICE
    {filterList          [0] IMPLICIT MonitorFilter,
     SEQUENCE
     {filterList         MonitorFilter          OPTIONAL,
      extensions         CSTAPrivateData       OPTIONAL} }

END -- of CSTA-change-monitor-filter
```

13.3 Monitor stop

```
CSTA-monitor-stop
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) monitor-stop( 103) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
MonitorCrossRefID FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

monitorStop      OPERATION
    ARGUMENT MonitorStopArgument
    RESULT      MonitorStopResult
    ERRORS      {universalFailure}
::= 73

MonitorStopArgument ::=
    CHOICE
    {crossRefIdentifier      MonitorCrossRefID,
     SEQUENCE
     {crossRefIdentifier      MonitorCrossRefID,
      extensions              CSTACommonArguments      OPTIONAL} }

MonitorStopResult ::=
    CHOICE
    {extensions              CSTAPrivateData,
     noData                  NULL}

END -- of CSTA-monitor-stop
```

13.4 Snapshot device

```
CSTA-snapshot-device
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) snapshot-device( 104) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
SnapshotDeviceData FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) }
DeviceID FROM Device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

snapshotDevice      OPERATION
    ARGUMENT SnapshotDeviceArgument
    RESULT      SnapshotDeviceResult
    ERRORS      {universalFailure}

::= 74

SnapshotDeviceArgument ::=
    CHOICE
    {snapshotObject      DeviceID,
     SEQUENCE
     {snapshotObject      DeviceID,
      extensions          CSTACommonArguments      OPTIONAL} }

SnapshotDeviceResult ::=
    CHOICE
    {snapshotData      SnapshotDeviceData,
     SEQUENCE
     {snapshotData      SnapshotDeviceData,
      extensions          CSTAPrivateData      OPTIONAL} }

END -- of CSTA-snapshot-device
```

13.5 Snapshot call

```
CSTA-snapshot-call
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) snapshot-call( 105) }
DEFINITIONS ::=
BEGIN
IMPORTS
OPERATION, ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
-- Data Types --
SnapshotCallData FROM CSTA-status-reporting
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) status-reporting( 126) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
CSTACommonArguments, CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) }
universalFailure FROM CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) };

snapshotCall      OPERATION
ARGUMENT          SnapshotCallArgument
RESULTS           SnapshotCallResult
ERRORS            {universalFailure}
::= 75

SnapshotCallArgument ::=
    CHOICE
    {snapshotObject      ConnectionID,
     SEQUENCE
     {snapshotObject      ConnectionID,
      extensions          CSTACommonArguments          OPTIONAL} }

SnapshotCallResult ::=
    CHOICE
    {snapshotData        SnapshotCallData,
     SEQUENCE
     {snapshotData        SnapshotCallData,
      extensions          CSTAPrivateData          OPTIONAL} }

END -- of CSTA-snapshot-call
```


14 Switching function errors

```
CSTA-error-definition
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) error-definition( 120) }

DEFINITIONS ::=
BEGIN
EXPORTS UniversalFailure, universalFailure;
IMPORTS ERROR FROM Remote-Operations-Notation
    { joint-iso-ccitt( 2) remote-operations( 4) notation( 0) }
CSTAPrivateData FROM CSTA-extension-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) extension-types( 129) };

universalFailure      ERROR
PARAMETER              UniversalFailure
::= 1

UniversalFailure ::= CHOICE
{ operationalErrors      [1] IMPLICIT Operations,
  stateErrors             [2] IMPLICIT StateIncompatibility,
  systemResourceErrors    [3] IMPLICIT SystemResourceAvailability,
  subscribedResourceAvailabilityErrors [4] IMPLICIT SubscribedResourceAvailability,
  performanceErrors       [5] IMPLICIT PerformanceManagement,
  securityErrors          [6] IMPLICIT SecurityError,
  unspecifiedErrors       [7] IMPLICIT NULL,
  nonStandardErrors       CSTAPrivateData}

Operations ::= ENUMERATED
{      generic              (1),
  requestIncompatibleWithObject (2),
  valueOutOfRange           (3),
  objectNotKnown            (4),
  invalidCallingDevice       (5),
  invalidCalledDevice        (6),
  invalidForwardingDestination (7),
  privilegeViolationOnSpecifiedDevice (8),
  privilegeViolationOnCalledDevice (9),
  privilegeViolationOnCallingDevice (10),
  invalidCSTACallIdentifier   (11),
  invalidCSTADeviceIdentifier (12),
  invalidCSTAConnectionIdentifier (13),
  invalidDestination         (14),
  invalidFeature             (15),
  invalidAllocationState     (16),
  invalidCrossRefID          (17),
  invalidObjectType          (18),
  securityViolation          (19) }

StateIncompatibility ::= ENUMERATED
{      generic              (1),
  invalidObjectState        (2),
  invalidConnectionID       (3),
```

	noActiveCall	(4),
	noHeldCall	(5),
	noCallToClear	(6),
	noConnectionToClear	(7),
	noCallToAnswer	(8),
	noCallToComplete	(9) }
SystemResourceAvailability ::= ENUMERATED		
{	generic	(1),
	serviceBusy	(2),
	resourceBusy	(3),
	resourceOutOfService	(4),
	networkBusy	(5),
	networkOutOfService	(6),
	overallMonitorLimitExceeded	(7),
	conferenceMemberLimitExceeded	(8) }
SubscribedResourceAvailability ::= ENUMERATED		
{	generic	(1),
	objectMonitorLimitExceeded	(2),
	externalTrunkLimitExceeded	(3),
	outstandingRequestLimitExceeded	(4) }
PerformanceManagement ::= ENUMERATED		
{	generic	(1),
	performanceLimitExceeded	(2) }
SecurityError ::= ENUMERATED		
{	unspecified	(0),
	sequenceNumberViolated	(1),
	timeStampViolated	(2),
	pACViolated	(3),
	sealViolated	(4) }
END -- of CSTA-error-definition		

15 **Switching event cause values**

```
CSTA-event-causes
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) event-causes( 121) }

DEFINITIONS ::=
BEGIN
EXPORTS  EventCause;
EventCause ::= ENUMERATED          -- a general list of cause codes
{
    activeMonitor          (1),
    alternate               (2),
    busy                   (3),
    callBack               (4),
    callCancelled          (5),
    callForwardAlways      (6),
    callForwardBusy        (7),
    callForwardNoAnswer    (8),
    callForward            (9),
    callNotAnswered        (10),
    callPickup             (11),
    campOn                 (12),
    destNotObtainable      (13),
    doNotDisturb           (14),
    incompatibleDestination (15),
    invalidAccountCode     (16),
    keyConference          (17),
    lockout                (18),
    maintenance            (19),
    networkCongestion      (20),
    networkNotObtainable   (21),
    newCall                (22),
    noAvailableAgents      (23),
    override               (24),
    park                   (25),
    overflow               (26),
    recall                 (27),
    redirected             (28),
    reorderTone            (29),
    resourcesNotAvailable  (30),
    silentMonitor          (31),
    transfer               (32),
    trunksBusy             (33),
    voiceUnitInitiator     (34) }

END -- of event-cause-definitions
```

16 CSTA data types

The major parameters have been assigned distinct application tags to facilitate parsing. The data is defined in logical groups in ascending order of application tag. Application tags used are:

- APPLICATION 1 - 5 : Device identifiers
- APPLICATION 11 - 14 : Connection identifiers and local connection states
- APPLICATION 21 - 24 : Status reporting
- APPLICATION 29 : CSTAPrivateData
- APPLICATION 30 : CSTACommonArguments

16.1 Switching function objects

```
CSTA-switching-function-objects
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) switching-function-objects( 122) }
DEFINITIONS ::=
BEGIN
EXPORTS CSTAObject;
IMPORTS
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) };

CSTAObject ::= CHOICE
    {device DeviceID,
     call ConnectionID}

END -- of CSTA-switching-function-objects
```

16.2 Device identifiers

```

CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
DEFINITIONS ::=
BEGIN
EXPORTS
DeviceID, NumberDigits, ExtendedDeviceID, CallingDeviceID, CalledDeviceID,
SubjectDeviceID, RedirectionDeviceID;

DeviceID ::= CHOICE
    { dialingNumber          [0] IMPLICIT   NumberDigits,
      deviceNumber          [1] IMPLICIT   DeviceNumber }

-- NumberDigits is a string of digits that represents a number (address) that
-- the switch can route a call on. It can be dialed by a user (i.e from a telephone
-- keypad) to make a call. DeviceNumber is not a routing address and cannot be dialed
-- from a keypad. It is allocated by the switch to reference a device.

NumberDigits ::= IA5String

DeviceNumber ::= INTEGER

ExtendedDeviceID ::= CHOICE
    { deviceIdentifier
      implicitPublic      [2]          IMPLICIT   DeviceID,
      explicitPublic      [3]          IMPLICIT   NumberDigits,
      implicitPrivate     [4]          IMPLICIT   PublicTON,
      explicitPrivate     [5]          IMPLICIT   NumberDigits,
      other               [6]          IMPLICIT   PrivateTON,
                                           OtherPlan }

CallingDeviceID ::= [APPLICATION 1] CHOICE
    { deviceIdentifier      ExtendedDeviceID,
      notKnown             [7]          IMPLICIT NULL,
      notRequired          [8]          IMPLICIT NULL }

CalledDeviceID ::= [APPLICATION 2] CHOICE
    { deviceIdentifier      ExtendedDeviceID,
      notKnown             [7]          IMPLICIT NULL,
      notRequired          [8]          IMPLICIT NULL }

SubjectDeviceID ::= [APPLICATION 3] CHOICE
    { deviceIdentifier      ExtendedDeviceID,
      notKnown             [7]          IMPLICIT NULL,
      notRequired          [8]          IMPLICIT NULL }

RedirectionDeviceID ::= [APPLICATION 4] CHOICE
    { numberDialed         ExtendedDeviceID,
      notKnown            [7]          IMPLICIT   NULL,
      notRequired         [8]          IMPLICIT   NULL }

-- SubjectDeviceID is used in some event reports to specify which device the report

```

-- refers to. If the SubjectDeviceID has had a monitor invoked then this data is not
-- required and so the implicit NULL encoding for notRequired is returned.
-- RedirectionDeviceID is used in Events as the lastRedirectionDevice.
-- CalledDeviceID is used in Events to specify the number dialled.

PublicTON ::= CHOICE

{	unknown	[0]	IMPLICIT IA5String,
	international	[1]	IMPLICIT IA5String,
	national	[2]	IMPLICIT IA5String,
	networkspecific	[3]	IMPLICIT IA5String,
	subscriber	[4]	IMPLICIT IA5String,
	abbreviated	[5]	IMPLICIT IA5String }

-- the public type of numbers are derived from CCITT E.164

PrivateTON ::= CHOICE

{	unknown	[0]	IMPLICIT IA5String,
	level3RegionalNumber	[1]	IMPLICIT IA5String,
	level2RegionalNumber	[2]	IMPLICIT IA5String,
	level1RegionalNumber	[3]	IMPLICIT IA5String,
	pTNSpecificNumber	[4]	IMPLICIT IA5String,
	localNumber	[5]	IMPLICIT IA5String,
	abbreviated	[6]	IMPLICIT IA5String }

-- the private type of numbers are derived from ECMA-155

OtherPlan ::= OCTET STRING -- allows future expansion to cover other numbering
-- plans (such as X.121 etc.)

END -- of CSTA-device-identifiers

16.3 Call and connection identifiers

```
CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
DEFINITIONS ::=
BEGIN
EXPORTS
ConnectionID;
IMPORTS
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

ConnectionID ::= [APPLICATION 11] IMPLICIT SEQUENCE
{
    call
    device CHOICE
        {staticID
         dynamicID
         DeviceID,
         [3] IMPLICIT OCTET STRING } OPTIONAL,
    [2] IMPLICIT OCTET STRING OPTIONAL,
    [3] IMPLICIT OCTET STRING } OPTIONAL}

END -- of CSTA-call-connection-identifiers
```


16.4 Connection states

```
CSTA-connection-states
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) connection-states( 125) }
DEFINITIONS ::=
BEGIN
EXPORTS
ConnectionList, ConnectionIDList, LocalConnectionState;
IMPORTS
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
DeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

ConnectionIDList ::= [APPLICATION 12] IMPLICIT SEQUENCE OF ConnectionID

CallInfo ::= [APPLICATION 13] IMPLICIT SEQUENCE OF SEQUENCE
    { endpoint          ConnectionID,
      staticEndpoint    DeviceID          OPTIONAL }

ConnectionList ::= CHOICE
    { connections      ConnectionIDList,
      callinformation   CallInfo }

LocalConnectionState ::= [APPLICATION 14] IMPLICIT ENUMERATED
    {
        null          (0),
        initiate       (1),
        alerting       (2),
        connect        (3),
        hold           (4),
        queued         (5),
        fail           (6) }

END -- CSTA-connection-states
```

16.5 Status reporting

```
CSTA-status-reporting
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) status-reporting( 126) }

DEFINITIONS ::=
BEGIN
EXPORTS
MonitorObject, MonitorCrossRefID, MonitorFilter, MonitorType, SnapshotCallData,
SnapshotDeviceData;
IMPORTS
DeviceID, SubjectDeviceID FROM CSTA-device-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) device-identifiers( 123) }
ConnectionID FROM CSTA-call-connection-identifiers
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
LocalConnectionState FROM CSTA-connection-states
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) connection-states( 125) }
CSTAObject FROM CSTA-switching-function-objects
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) switching-function-objects( 122) };

MonitorObject ::= CSTAObject

MonitorCrossRefID ::= [APPLICATION 21] IMPLICIT OCTET STRING

MonitorFilter      ::= SEQUENCE      -- default is no filter (i.e. all events)
{
  call      [0] IMPLICIT CallFilter DEFAULT {}
  feature   [1] IMPLICIT FeatureFilter      DEFAULT {}
  agent     [2] IMPLICIT AgentFilter        DEFAULT {}
  maintenance [3] IMPLICIT MaintenanceFilter DEFAULT {}
  private   [4] IMPLICIT BOOLEAN            DEFAULT FALSE }

-- setting the relevant bit requests the filter for the appropriate events

CallFilter ::= BIT STRING
{
  callCleared      (0),
  conferenced      (1),
  connectionCleared (2),
  delivered         (3),
  diverted          (4),
  established       (5),
  failed            (6),
  held              (7),
  networkReached    (8),
  originated        (9),
  queued            (10),
  retrieved         (11),
  serviceInitiated  (12),
  transferred       (13) }
```

```
FeatureFilter ::= BIT STRING
{
    callInformation          (0),
    doNotDisturb             (1),
    forwarding               (2),
    messageWaiting           (3) }

AgentFilter ::= BIT STRING
{
    loggedIn                (0),
    loggedOff               (1),
    notReady                (2),
    ready                   (3),
    workNotReady            (4),
    workReady               (5) }

MaintenanceFilter ::= BIT STRING
{
    backInService           (0),
    outOfService            (1) }

MonitorType ::= ENUMERATED
{
    call                    (0),
    device                  (1) }

SnapshotDeviceData ::= [APPLICATION 22] IMPLICIT SEQUENCE OF
    SnapshotDeviceResponseInfo

SnapshotDeviceResponseInfo ::= SEQUENCE
{callIdentifier            ConnectionID,
 localCallState            CallState }

SnapshotCallData ::= [APPLICATION 23] IMPLICIT SEQUENCE OF
    SnapshotCallResponseInfo

SnapshotCallResponseInfo ::= SEQUENCE
{deviceOnCall              SubjectDeviceID,
 callIdentifier            ConnectionID,
 localConnectionState      LocalConnectionState OPTIONAL }

CallState ::= CHOICE
{ compound                 [0] IMPLICIT CompoundCallState,
  simple                   [1] IMPLICIT SimpleCallState,
  unknown                  [2] IMPLICIT NULL }

-- unknown is returned by server if no other CallState can be supplied --

CompoundCallState ::= SEQUENCE OF LocalConnectionState

SimpleCallState ::= ENUMERATED
{callNull                  (0), -- '00'H - null-null
 callPending               (1), -- '01'H - null-initiate
 callOriginated            (3), -- '03'H - null-connect
 callDelivered             (35), -- '23'H - alerting-connect
 callDeliveredHeld         (36), -- '24'H - alerting-held
```

callReceived	(50),	-- '32'H	- connect-alerting
callEstablished	(51),	-- '33'H	- connect-connect
callEstablishedHeld	(52),	-- '34'H	- connected-held
callReceivedOnHold	(66),	-- '42'H	- held-alerting
callEstablishedOnHold	(67),	-- '43'H	- held-connect
callQueued	(83),	-- '53'H	- queued-connect
callQueuedHeld	(84),	-- '54'H	- queued-held
callFailed	(99),	-- '63'H	- failed-connect
callFailedHeld	(100)}	-- '64'H	- failed-held

-- This represents the main call states in a simplified encoding. The semantics
-- are identical to the sequence of connection states but they are represented by
-- an item from an enumerated list.

END -- of CSTA-status-reporting

16.6 Device and feature types and other parameters

```
CSTA-device-feature-types
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-feature-types( 127) }
DEFINITIONS ::=
BEGIN
EXPORTS
AccountInfo, AgentID, AgentGroup, AgentPassword, AgentParameter, AgentState,
AllocationState, AuthCode, ConnectionDetails, DeviceClass, DeviceInfo, DeviceType,
DivertInfo, FeatureInfo, ListForwardParameters, LoggedOnInfo, LoggedOffInfo,
ForwardParameter, ForwardingType, NoOfCallsInQueue, QueryDeviceFeature,
QueryDeviceInformation, ReserveConnection, SetDeviceFeature, SystemStatus, SelectValue,
PriorityValue, SetUpValues, RetryValue, RouteUsedFlag, DomainValue, RoutingCrossRefID;
IMPORTS
ConnectionID FROM CSTA-call-connection-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) call-connection-identifiers( 124) }
DeviceID, NumberDigits, CallDeviceID FROM CSTA-device-identifiers
    { iso( 1) identified-organization( 3) icd-ecma( 0012)
      standard( 0) csta( 180) version1( 1) device-identifiers( 123) };

AccountInfo      ::= OCTET STRING  -- used in CallInformation event

AgentID ::= OCTET STRING

AgentGroup      ::= DeviceID

AgentPassword    ::= OCTET STRING

AgentParameter   ::= CHOICE
{
    loggedIn           [0]          IMPLICIT LoggedOnInfo,
    loggedOut          [1]          IMPLICIT LoggedOffInfo,
    notReady           [2]          IMPLICIT NULL,
    ready              [3]          IMPLICIT NULL,
    workNotReady       [4]          IMPLICIT NULL,
    workReady          [5]          IMPLICIT NULL }

AgentState       ::= ENUMERATED
{
    notReady           (0),
    null               (1),
    ready              (2),
    workNotReady       (3),
    workReady          (4) }

AllocationState  ::= ENUMERATED
{
    callDelivered      (0),
    callEstablished    (1) }
-- used in MakePredictiveCall to indicate when call should be allocated to the device

AuthCode        ::= OCTET STRING  -- used in Information Events
```

ConnectionDetails ::= CHOICE

{	heldCall	[0] IMPLICIT	ConnectionID,
	activeCall	[1] IMPLICIT	ConnectionID,
	bothCalls	[2] IMPLICIT	SEQUENCE
			{heldCall ConnectionID,
			activeCall ConnectionID} }

DeviceClass ::= BIT STRING

{	voice	(0),
	data	(1),
	image	(2),
	other	(3) }

DeviceInfo ::= SEQUENCE

{	deviceId	DeviceID	OPTIONAL,
	deviceType	DeviceType	OPTIONAL,
	deviceClass	DeviceClass	OPTIONAL}

DeviceType ::= ENUMERATED

{	station	(0),
	line	(1),
	button	(2),
	aCD	(3),
	trunk	(4),
	operator	(5),
	station-group	(16),
	line-group	(17),
	button-group	(18),
	aCD-group	(19),
	trunk-group	(20),
	operator-group	(21),
	other	(255) }

DivertInfo ::= CHOICE --used by Divert Call service

{	deflect	[0] IMPLICIT	SEQUENCE
			{ callToBeDiverted ConnectionID,
			newDestination CalledDeviceID }
	pickup	[1] IMPLICIT	SEQUENCE
			{ callToBePickedUp ConnectionID,
			requestingDevice DeviceID }
	group	[2]	DeviceID }

FeatureInfo ::= CHOICE --used by Call Completion service

{	campon	[0] IMPLICIT	ConnectionID,
	callback	[1] IMPLICIT	ConnectionID,
	intrude	[2] IMPLICIT	ConnectionID }

ListForwardParameters ::= SEQUENCE OF SEQUENCE

{	forwardingType	ForwardingType,
	forwardDN	NumberDigits }

LoggedOnInfo ::= SEQUENCE

{	agentID	[10] IMPLICIT	AgentID	OPTIONAL,
	password	[11] IMPLICIT	AgentPassword	OPTIONAL,
	group		AgentGroup	OPTIONAL }

LoggedOffInfo	::= SEQUENCE		
{	agentID	[10] IMPLICIT	AgentID
	group		AgentGroup
			OPTIONAL, OPTIONAL }
ForwardParameter	::= SEQUENCE		
{	forwardingType		ForwardingType,
	forwardDN		NumberDigits
			OPTIONAL }
ForwardingType	::= ENUMERATED		
{	forwardImmediateOn	(0),	
	forwardImmediateOff	(1),	
	forwardBusyOn	(2),	
	forwardBusyOff	(3),	
	forwardNoAnsOn	(4),	
	forwardNoAnsOff	(5),	
	forwardBusyIntOn	(6),	
	forwardBusyIntOff	(7),	
	forwardBusyExtOn	(8),	
	forwardBusyExtOff	(9),	
	forwardNoAnsIntOn	(10),	
	forwardNoAnsIntOf	(11),	
	forwardNoAnsExtOn	(12),	
	forwardNoAnsExtOff	(13) }	
NoOfCallsInQueue	::= INTEGER		-- used in Call Queued Event
QueryDeviceFeature	::=ENUMERATED		-- used by Query Feature service request
{	msgWaitingOn	(0),	
	doNotDisturbOn	(1),	
	forward	(2),	
	lastDialedNumber	(3),	
	deviceInfo	(4),	
	agentState	(5) }	
QueryDeviceInformation	::= CHOICE		
{	msgWaitingOn	[0]	IMPLICIT BOOLEAN,
	doNotDisturbOn	[1]	IMPLICIT BOOLEAN,
	forward	[2]	IMPLICIT ListForwardParameters,
	lastDialed	[3]	IMPLICIT NumberDigits,
	deviceInfo	[4]	IMPLICIT DeviceInfo,
	agentState	[5]	IMPLICIT AgentState }
ReserveConnection	::= BOOLEAN		-- used with Hold service to reserve ISDN -- connection
SetDeviceFeature	::= CHOICE		-- used by SetFeature service request
{	msgWaitingOn	[0]	IMPLICIT BOOLEAN,
	doNotDisturbOn	[1]	IMPLICIT BOOLEAN,
	forward	[2]	IMPLICIT ForwardParameter,
	aRequestedAgentState	[3]	AgentParameter }

SystemStatus ::= ENUMERATED

{	initializing	(0),
	enabled	(1),
	normal	(2),
	messagesLost	(3),
	disabled	(4),
	overloadImminent	(5),
	overloadReached	(6),
	overloadRelieved	(7) }

SelectValue ::= ENUMERATED

{	normal	(0),
	leastCost	(1),
	emergency	(2),
	aCD	(3),
	userDefined	(4) }

PriorityValue ::= BOOLEAN

-- TRUE means priority call

SetUpValues ::= OCTET STRING

-- Contains Q.931 Setup message

RetryValue ::= CHOICE

-- used in RouteSelect Request service

{	noListAvailable	[0] IMPLICIT BOOLEAN,
	noCountAvailable	[1] IMPLICIT BOOLEAN,
	retryCount	[2] IMPLICIT INTEGER }

RouteUsedFlag ::= BOOLEAN

-- TRUE means RouteUsed Request service

-- requested

DomainValue ::= BOOLEAN

-- TRUE means CSTA subdomain destination

RoutingCrossRefID ::= [APPLICATION 24] IMPLICIT OCTET STRING

END -- of CSTA-device-feature-types

16.7 Security service

```
CSTA-security
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) security( 128) }

DEFINITIONS ::=
BEGIN
EXPORTS
CSTASecurityData;
IMPORTS
PrivilegeAttributeCertificate FROM Security-Information
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) desd( 138) securityData( 1) };

Seal ::= SEQUENCE
{
  algorithmIdentifier      OBJECT IDENTIFIER,
  keyIdentifier            OCTET STRING,
  seal                    OCTET STRING }

CSTASecurityData ::= SEQUENCE
{
  messageSequenceNumber    INTEGER      OPTIONAL,
  timeStamp                UTCTime      OPTIONAL,
  privilegeAttributeCertificate PrivilegeAttributeCertificate
  seal                    Seal          OPTIONAL }

END -- of CSTA-security
```

16.8 Common extensions

```
CSTA-extension-types
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) extension-types( 129) }
DEFINITIONS ::=
BEGIN
EXPORTS
CSTACommonArguments, CSTAPrivateData;
IMPORTS
CSTASecurityData FROM CSTA-security
{ iso( 1) identified-organization( 3) icd-ecma( 0012)
  standard( 0) csta( 180) version1( 1) security( 128) };

CSTACommonArguments ::= [APPLICATION 30] IMPLICIT SEQUENCE
{
  security                [0] IMPLICIT CSTASecurityData    OPTIONAL
  privateData             [1] IMPLICIT SEQUENCE OF CSTAPrivateData
                           OPTIONAL }

CSTAPrivateData ::= [APPLICATION 29] IMPLICIT SEQUENCE
{
  manufacturerOBJECT IDENTIFIER,
  ANY DEFINED BY          manufacturer}

-- Manufacturer specific (or standard) extensions shall be uniquely identified using
-- Object Identifiers issued by ECMA according to ISO 6523.

END -- of CSTA-extension-types
```

Annex A
(normative)

Protocol Implementation Conformance Statement (PICS) Proforma

Contents

A.1	Introduction	83
A.2	Definitions and abbreviations	83
A.3	Conformance	83
A.4	Instructions for completing the PICS proforma	83
A.5	Implementation Identification	85
A.6	Switching Function Services	86
A.6.1	Alternate Call	86
A.6.2	Answer Call	86
A.6.3	Call Completion	87
A.6.4	Clear Call	87
A.6.5	Clear Connection	88
A.6.6	Conference Call	88
A.6.7	Consultation Call	89
A.6.8	Divert Call	90
A.6.9	Hold Call	91
A.6.10	Make Call	91
A.6.11	Make Predictive Call	92
A.6.12	Query Device	93
A.6.13	Reconnect Call	94
A.6.14	Retrieve Call	95
A.6.15	Set Feature	96
A.6.16	Transfer Call	97
A.7	Switching Function Events	98
A.7.1	Call Events	98
A.7.1.1	Event Macro	98
A.7.1.2	Call Cleared	98
A.7.1.3	Conferenced	99
A.7.1.4	Connection Cleared	99
A.7.1.5	Delivered	100
A.7.1.6	Diverted	100
A.7.1.7	Established	101
A.7.1.8	Failed	101
A.7.1.9	Held	102
A.7.1.10	Network Reached	102

	A.7.1.11	Originated	103
	A.7.1.12	Queued	103
	A.7.1.13	Retrieved	104
	A.7.1.14	Service Initiated	104
	A.7.1.15	Transferred	105
A.7.2		Agent Events	106
	A.7.2.1	Logged On	106
	A.7.2.2	Logged Off	106
	A.7.2.3	Ready	106
	A.7.2.4	Not Ready	107
	A.7.2.5	Work Ready	107
	A.7.2.6	Work Not Ready	107
A.7.3		Other Feature Events	108
	A.7.3.1	Call Information	108
	A.7.3.2	Do Not Disturb	108
	A.7.3.3	Forwarding	109
	A.7.3.4	Message Waiting	110
A.7.4		Maintenance Events	110
	A.7.4.1	Back In Service	110
	A.7.4.2	Out of Service	110
A.7.5		Private Events	111
	A.7.5.1	Private	111
A.8		Computing Services	111
	A.8.1	Route Request	111
	A.8.2	Re-Route Service	112
	A.8.3	Route Select	112
	A.8.4	Route Used	113
	A.8.5	Route End	113
A.9		Bidirectional Services	114
	A.9.1	Escape Service	114
	A.9.2	System Status	115
A.10		Status Reporting Services	116
	A.10.1	Change Monitor Filter	116
	A.10.2	Monitor Start	117
	A.10.3	Monitor Stop	118
	A.10.4	Snapshot Call	118
	A.10.5	Snapshot Device	119
A.11		Switching Event Cause Values	120
A.12		Switching Function Errors	122
	A.12.1	Operational Errors	122
	A.12.2	State Incompatibility	123

A.12.3	System Resource Availability Errors	123
A.12.4	Subscribed Resource Availability Errors	124
A.12.5	Performance Errors	124
A.13	CSTA Data Types	125
A.14	Security	125

A.1 Introduction

The Protocol Implementation Conformance Statement (PICS) is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use:

- by the protocol implementor, as a check-list to reduce the risk of failure to conform to the standard through oversight;
- by the supplier and acquirer (or potential acquirer) of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- by the user (or potential user) of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking cannot be guaranteed, failure to interwork can often be predicted from incompatible PICS);
- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Definitions and abbreviations

This Standard uses the following terms defined in ISO 9646-1:

- Protocol Implementation Conformance Statement (PICS);
- PICS Proforma.

In the "Reference" columns of the body of the PICS proforma, the letter S refers to the CSTA Services standard ECMA-179 and the letter P refers to the CSTA Protocol standard ECMA-180.

The following terms are used in the "Status" columns of the body of the PICS proforma:

m = mandatory; the capability is required for conformance to the protocol.

o = optional; the capability is not required for conformance to the protocol, or is required only within constraints described in dependencies ("if" statements). If the capability is implemented, it is required to conform to the protocol specifications.

o.<n> = optional, but support of at least one of the group of options labelled by the same numeral <n> is required.

C.<cid> = conditional; the requirement is conditional according to the condition identified by <cid>.

<item> = simple-predicate condition, dependent on the support marked for <item>.

A.3 Conformance

The supplier of a protocol implementation which is claimed to conform to ECMA-180 shall complete a copy of the Protocol Implementation Conformance Statement (PICS) proforma in clauses A.5 to A.14.

A.4 Instructions for completing the PICS proforma

The first part of the PICS proforma, the Implementation Identification (clause A.5), is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma (clauses A.6 to A.14) is a fixed format questionnaire divided into subclauses each containing a group of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by marking an answer to indicate a restricted choice (usually Yes or No), or by checking off all supported values (for parameters with a default).

Each item is identified by an item reference in the first column; the second column title indicates the nature of the table items which follow. The third column contains the references to material that specifies the item in the main body of ECMA-179 and ECMA-180. The remaining columns record the status of the item - whether support is mandatory, optional, or not applicable - and provide space for the answers.

Where a service is not supported, any parameters or dependent service components are not applicable. These dependencies are indicated in the status column using the item identifier as a key. It is not necessary to complete items in any subsidiary sections if a "No" response is given to the primary service component.

For supported services, a negative response to a mandatory subsidiary item indicates that the service does not conform to ECMA-180, and conformance cannot be claimed for that service.

A.5 Implementation Identification

Supplier	
Protocol Version	First Edition
Date of Statement	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification - e.g. name(s) and version(s) for machines and/or operating systems; system name(s)	

Note A.1

The first five items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification.

Note A.2

The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g. Type, Series, Model).

A.6 Switching Function Services

A.6.1 Alternate Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A1	Alternate Call service	S9.1 P9.1	o		Yes [] No []
A1a	Service Result	P5.3 P9.1	A1:m	[]	Yes []
A1b	Held Call parameter	P9.1	A1:o1		Yes [] No []
A1c	Active Call parameter	P9.1	A1:o1		Yes [] No []
A1d	Security parameters	P16.8	A1:o		Yes [] No []
A1e	Private Data in Request	P16.8	A1:o		Yes [] No []
A1f	Private Data in Result	P16.8	A1:o		Yes [] No []
A1g	Report of Service Errors	P14	A1:m	[]	Yes []

A.6.2 Answer Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A2	Answer Call service	S9.2 P9.2	o		Yes [] No []
A2a	Service Result	P5.3 P9.2	A2:m	[]	Yes []
A2b	Call to be Answered parameter	P9.2	A2:m	[]	Yes []
A2c	Security parameters	P16.8	A2:o		Yes [] No []
A2d	Private Data in Request	P16.8	A2:o		Yes [] No []
A2e	Private Data in Result	P16.8	A2:o		Yes [] No []
A2f	Report of Service Errors	P14	A2:m	[]	Yes []

A.6.3 Call Completion

Item	Service / Feature	Reference	Status	N/A	Supported?
A3	Call Completion Service	S9.3 P9.3	o		Yes [] No []
A3a	Camp On feature	P9.3	A3:o1		Yes [] No []
A3b	Call Back feature	P9.3	A3:o1		Yes [] No []
A3c	Intrude feature	P9.3	A3:o1		Yes [] No []
A3d	Service Result	P5.3 P9.3	A3:m	[]	Yes []
A3e	Call to Complete parameter	P9.3	A3:m	[]	Yes []
A3f	Security parameters	P16.8	A3:o		Yes [] No []
A3g	Private Data in Request	P16.8	A3:o		Yes [] No []
A3h	Private Data in Result	P16.8	A3:o		Yes [] No []
A3i	Report of Service Errors	P14	A3:m	[]	Yes []

A.6.4 Clear Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A4	Clear Call service	S9.4 P9.4	o		Yes [] No []
A4a	Service Result	P5.3 P9.4	A4:m	[]	Yes []
A4b	Call to be Cleared parameter	P9.4	A4:m	[]	Yes []
A4c	Security parameters	P16.8	A4:o		Yes [] No []
A4d	Private Data in Request	P16.8	A4:o		Yes [] No []
A4e	Private Data in Result	P16.8	A4:o		Yes [] No []
A4f	Report of Service Errors	P14	A4:m	[]	Yes []

A.6.5 Clear Connection

Item	Service / Feature	Reference	Status	N/A	Supported?
A5	Clear Connection service	S9.5 P9.5	o		Yes [] No []
A5a	Service Result	P5.3 P9.5	A5:m	[]	Yes []
A5b	Connection to be Cleared parameter	P9.5	A5:m	[]	Yes []
A5c	Security parameters	P16.8	A5:o		Yes [] No []
A5d	Private Data in Request	P16.8	A5:o		Yes [] No []
A5e	Private Data in Result	P16.8	A5:o		Yes [] No []
A5f	Report of Service Errors	P14	A5:m	[]	Yes []

A.6.6 Conference Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A6	Conference Call service	S9.6 P9.6	o		Yes [] No []
A6a	Service Result	P5.3 P9.6	A6:m	[]	Yes []
A6b	Held Call parameter	P9.6	A6:o1		Yes [] No []
A6c	Active Call parameter	P9.6	A6:o1		Yes [] No []
A6d	Conference Call parameter in Result	P9.6	A6a:m	[]	Yes []
A6e	Connection ID list	P9.6	A6:o		Yes [] No []
A6f	Static Device ID included in list	P16.4	A6e:o	[]	Yes [] No []
A6g	Security parameters	P16.8	A6:o		Yes [] No []
A6h	Private Data in Request	P16.8	A6:o		Yes [] No []
A6i	Private Data in Result	P16.8	A6:o		Yes [] No []
A6j	Report of Service Errors	P14	A6:m	[]	Yes []

A.6.7 Consultation Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A7	Consultation Call service	S9.7 P9.7	o		Yes [] No []
A7a	Service Result	P5.3 P9.7	A7:m	[]	Yes []
A7b	Existing call parameter	P9.7	A7:m	[]	Yes []
A7c	Called Device ID parameter	P9.7	A7:m	[]	Yes []
A7d	Initiated Call parameters in Result	P9.7	A7a:m	[]	Yes []
A7e	Security parameters	P16.8	A7:o		Yes [] No []
A7f	Private Data in Request	P16.8	A7:o		Yes [] No []
A7g	Private Data in Result	P16.8	A7:o		Yes [] No []
A7h	Report of Service Errors	P14	A7:m	[]	Yes []

A.6.8 Divert Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A8	Divert Call service	S9.8 P9.8	o		Yes [] No []
A8a	Deflect feature	P9.8	A8:o1		Yes [] No []
A8b	Directed Pickup feature	P9.8	A8:o1		Yes [] No []
A8c	Group Pickup feature	P9.8	A8:o1		Yes [] No []
A8d	Service Result	P5.3 P9.8	A8:m	[]	Yes []
A8e	Call to be Diverted parameter	P16.6	A8a:m	[]	Yes []
A8f	New Destination parameter	P16.6	A8a:m	[]	Yes []
A8g	Call to be Picked Up parameter	P16.6	A8b:m	[]	Yes []
A8h	Requesting Device parameter	P16.6	A8b:o	[]	Yes []
A8i	Device ID of group parameter	P16.6	A8c:m	[]	Yes []
A8j	Security parameters	P16.8	A8:o		Yes [] No []
A8k	Private Data in Request	P16.8	A8:o		Yes [] No []
A8l	Private Data in Result	P16.8	A8:o		Yes [] No []
A8m	Report of Service Errors	P14	A8:m	[]	Yes []

A.6.9 Hold Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A9	Hold Call service	S9.9 P9.9	o		Yes [] No []
A9a	Service Result	P5.3 P9.9	A9:m	[]	Yes []
A9b	Call to be Held	P9.9	A9:m	[]	Yes []
A9c	Procedures for Connection Reservation parameter = TRUE	S9.9.1	A9:o		Yes [] No []
A9d	Security parameters	P16.8	A9:o		Yes [] No []
A9e	Private Data in Request	P16.8	A9:o		Yes [] No []
A9f	Private Data in Result	P16.8	A9:o		Yes [] No []
A9g	Report of Service Errors	P14	A9:m	[]	Yes []

A.6.10 Make Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A10	Make Call service	S9.10 P9.10	o		Yes [] No []
A10a	Service Result	P5.3 P9.10	A10:m	[]	Yes []
A10b	Calling Device ID parameter	P9.10	A10:m	[]	Yes []
A10c	Called Device ID parameter	P9.10	A10:m	[]	Yes []
A10d	Initiated Call parameter in Result	P9.10	A10a:m	[]	Yes []
A10e	Security parameters	P16.8	A10:o		Yes [] No []
A10f	Private Data in Request	P16.8	A10:o		Yes [] No []
A10g	Private Data in Result	P16.8	A10:o		Yes [] No []
A10h	Report of Service Errors	P14	A10:m	[]	Yes []

A.6.11 Make Predictive Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A11	Make Predictive service	S9.11 P9.11	o		Yes [] No []
A11a	Service Result	P5.3 P9.11	A11:m	[]	Yes []
A11b	Calling Device ID parameter	P9.11	A11:m	[]	Yes [] No []
A11c	Called Device ID parameter	P9.11	A11:m	[]	Yes [] No []
A11d	Allocation on Established condition	S9.11.1	A11:o		Yes [] No []
A11e	Initiated Call parameter in Result	P9.11	A11a:m	[]	Yes []
A11f	Security parameters	P16.8	A11:o		Yes [] No []
A11g	Private Data in Request	P16.8	A11:o		Yes [] No []
A11h	Private Data in Result	P16.8	A11:o		Yes [] No []
A11i	Report of Service Errors	P14	A11:m	[]	Yes []

A.6.12 Query Device

Item	Service / Feature	Reference	Status	N/A	Supported?
A12	Query Device service	S9.12 P9.12	o		Yes [] No []
A12a	Message Waiting feature	P9.12	A12:o1		Yes [] No []
A12b	Do Not Disturb feature	P9.12	A12:o1		Yes [] No []
A12c	Forwarding feature	P9.12	A12:o1		Yes [] No []
A12d	Last Number feature	P9.12	A12:o1		Yes [] No []
A12e	Device Info feature	P9.12	A12:o1		Yes [] No []
A12f	Agent State feature	P9.12	A12:o1		Yes [] No []
A12g	Service Result	P5.3 P9.12	A12:m	[]	Yes []
A12h	Device parameter in Request	P9.12	A12:m	[]	Yes []
A12i	Feature parameter in Request	P9.12	A12:m	[]	Yes []
A12j	Device Information in Service Result	P9.12	A12g:m	[]	Yes []
A12k	Message Waiting On	P16.6	A12a:m	[]	Yes [] No []
A12l	Do Not Disturb On	P16.6	A12b:m	[]	Yes [] No []
A12m	Forward Immediate	P16.6	A12c:o1	[]	Yes [] No []
A12n	Forward Busy	P16.6	A12c:o1	[]	Yes [] No []
A12o	Forward No Answer	P16.6	A12c:o1	[]	Yes [] No []
A12p	Forward Busy Internal	P16.6	A12c:o1	[]	Yes [] No []
A12q	Forward Busy External	P16.6	A12c:o1	[]	Yes [] No []
A12r	Forward No Answer Internal	P16.6	A12c:o1	[]	Yes [] No []
A12s	Forward No Answer External	P16.6	A12c:o1	[]	Yes [] No []
A12t	Forward-to Number	P16.6	A12c:m	[]	Yes []
A12u	Last Dialed Number	P16.6	A12d:m	[]	Yes []
A12v	Device ID	P16.6	A12e:o1	[]	Yes [] No []
A12w	Device Type	P16.6	A12e:o1	[]	Yes [] No []
A12x	Device Class	P16.6	A12e:o1	[]	Yes [] No []

A.6.12 Query Device (continued)

Item	Service / Feature	Reference	Status	N/A	Supported?
A12y	Null	P16.6	A12f:o1	<input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/>
A12z	Not Ready	P16.6	A12f:o1	<input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/>
A12aa	Ready	P16.6	A12f:o1	<input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/>
A12bb	Work Not Ready	P16.6	A12f:o1	<input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/>
A12cc	Work Ready	P16.6	A12f:o1	<input type="checkbox"/>	Yes <input type="checkbox"/> No <input type="checkbox"/>
A12dd	Security parameters	P16.8	A12:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A12ee	Private Data in Request	P16.8	A12:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A12ff	Private Data in Result	P16.8	A12:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A12gg	Report of Service Errors	P14	A12:m	<input type="checkbox"/>	Yes <input type="checkbox"/>

A.6.13 Reconnect Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A13	Reconnect Call service	S9.13 P9.13	o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13a	Service Result	P5.3 P9.13	A13:m	<input type="checkbox"/>	Yes <input type="checkbox"/>
A13b	Held Call parameter	P9.13	A13:01		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13c	Active Call parameter	P9.13	A13:01		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13d	Security parameters	P16.8	A13:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13e	Private Data in Request	P16.8	A13:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13f	Private Data in Result	P16.8	A13:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
A13g	Report of Service Errors	P14	A13:m	<input type="checkbox"/>	Yes <input type="checkbox"/>

A.6.14 Retrieve Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A14	Retrieve Call service	S9.14 P9.14	o		Yes [] No []
A14a	Service Result	P5.3 P9.14	A14:m	[]	Yes []
A14b	Call to be Retrieved parameter	P9.14	A14:m	[]	Yes []
A14c	Security parameters	P16.8	A14:o		Yes [] No []
A14d	Private Data in Request	P16.8	A14:o		Yes [] No []
A14e	Private Data in Result	P16.8	A14:o		Yes [] No []
A14f	Report of Service Errors	P14	A14:m	[]	Yes []

A.6.15 Set Feature

Item	Service / Feature	Reference	Status	N/A	Supported?
A15	Set Feature service	S9.15 P9.15	o		Yes [] No []
A15a	Message Waiting feature	P9.15	A15:o1		Yes [] No []
A15b	Do Not Disturb feature	P9.15	A15:o1		Yes [] No []
A15c	Forwarding feature	P9.15	A15:o1		Yes [] No []
A15d	Agent Parameter feature	P9.15	A15:o1		Yes [] No []
A15e	Service Result	P5.3 P9.15	A15:m	[]	Yes []
A15f	Device parameter in Request	P9.15	A15:m	[]	Yes []
A15g	Feature parameter in Request	P9.15	A15:m	[]	Yes []
A15h	Forward Always	P9.15	A15c:01	[]	Yes [] No []
A15i	Forward Busy	P9.15	A15c:o1	[]	Yes [] No []
A15j	Forward No Answer	P9.15	A15c:o1	[]	Yes [] No []
A15k	Forward Busy Internal	P9.15	A15c:o1	[]	Yes [] No []
A15l	Forward Busy External	P9.15	A15c:o1	[]	Yes [] No []
A15m	Forward No Answer Internal	P9.15	A15c:o1	[]	Yes [] No []
A15n	Forward No Answer External	P9.15	A15c:o1	[]	Yes [] No []
A15o	Forward to Device	P9.15	A15c:o	[]	Yes [] No []
A15p	Login	P9.15	A15d:o1	[]	Yes [] No []
A15q	Logout	P9.15	A15d:o1	[]	Yes [] No []
A15r	Ready	P9.15	A15d:o1	[]	Yes [] No []
A15s	Not Ready	P9.15	A15d:o1	[]	Yes [] No []
A15t	Work Not Ready	P9.15	A15d:01	[]	Yes [] No []
A15u	Work Ready	P9.15	A15d:o1	[]	Yes [] No []
A15v	Agent ID	P9.15	c1:o	[]	Yes [] No []
A15w	ACD Pilot or Group	P9.15	c1:o	[]	Yes [] No []
A15x	Agent Password	P9.15	A15q:o	[]	Yes [] No []
A15y	Security parameters	P16.8	A15:o		Yes [] No []

A.6.15 Set Feature (continued)

Item	Service / Feature	Reference	Status	N/A	Supported?
A15z	Private Data in Request	P16.8	A15:o		Yes [] No []
A12aa	Private Data in Result	P16.8	A15:o		Yes [] No []
A12bb	Report of Service Errors	P14	A15:m	[]	Yes []

Note A.3

c1: (A15p or A15q)

A.6.16 Transfer Call

Item	Service / Feature	Reference	Status	N/A	Supported?
A16	Transfer Call service	S9.16 P9.16	o		Yes [] No []
A16a	Service Result	P5.3 P9.16	A16:m	[]	Yes []
A16b	Held Call parameter	P9.16	A16:m	[]	Yes []
A16c	Active Call parameter	P9.16	A16:m	[]	Yes []
A16d	Transferred Call parameter in Result	P9.16	A16:o		Yes [] No []
A16e	List of remaining parties	P9.16	A16:o		Yes [] No []
A16f	Static IDs included in list	P16.4	A16e:o	[]	Yes [] No []
A16g	Security parameters	P16.8	A16:o		Yes [] No []
A16h	Private Data in Request	P16.8	A16:o		Yes [] No []
A16i	Private Data in Result	P16.8	A16:o		Yes [] No []
A16j	Report of Service Errors	P14	A16:m	[]	Yes []

A.7 Switching Function Events

A.7.1 Call Events

A.7.1.1 Event Macro

Item	Service / Feature	Reference	Status	N/A	Supported?
B1	Event macro	S10.2 P10	o		Yes [] No []
B1a	Monitor CrossRefID	S10.2.1 P5.4 P10	B1:m	[]	Yes []
B1b	EventTypeID	P10	B1:m	[]	Yes []
B1c	EventInfo	P10	B1:m	[]	Yes []
B1d	CSTA Private Data	P10	B1:o		Yes [] No []

A.7.1.2 Call Cleared

Item	Service / Feature	Reference	Status	N/A	Supported?
B2	Call Cleared event	S10.2.3.1 P10.1.1	o		Yes [] No []
B2a	Cleared Call parameter	P10.1.1	B2:m	[]	Yes []
B2b	Cause parameter	P10.1.1	B2:o		Yes [] No []
B2c	Local Connection information	P10.1.1	B2:o		Yes [] No []

A.7.1.3 Conferenced

Item	Service / Feature	Reference	Status	N/A	Supported?
B3	Conferenced event	S10.2.3.2 P10.1.2	o		Yes [] No []
B3a	Primary Old Call parameter	P10.1.2	B3:m	[]	Yes []
B3b	Secondary Old Call parameter	P10.1.2	B3:C.1	[]	Yes [] No []
B3c	Conference Controller parameter	P10.1.2	B3:m	[]	Yes []
B3d	Added Party parameter	P10.1.2	B3:m	[]	Yes []
B3e	Cause parameter	P10.1.2	B3:o		Yes [] No []
B3f	Local Connection information	P10.1.2	B3:o		Yes [] No []
B3g	Connection ID list	P10.1.2	B3:o		Yes [] No []
B3h	Static Device ID included in list	P16.4	B3g:o	[]	Yes [] No []

C.1: If provided in previous events then mandatory
else optional

A.7.1.4 Connection Cleared

Item	Service / Feature	Reference	Status	N/A	Supported?
B4	Connection Cleared event	S10.2.3.3 P10.1.3	o		Yes [] No []
B4a	Dropped Connection parameter	P10.1.3	B4:m	[]	Yes []
B4b	Releasing Device parameter	P10.1.3	B4:m	[]	Yes []
B4c	Cause parameter	P10.1.3	B4:o		Yes [] No []
B4d	Local Connection information	P10.1.3	B4:o		Yes [] No []

A.7.1.5 Delivered

Item	Service / Feature	Reference	Status	N/A	Supported?
B5	Delivered event	S10.2.3.4 P10.1.4	o		Yes [] No []
B5a	Alerting Connection ID parameter	P10.1.4	B5:m	[]	Yes []
B5b	Alerting Device ID parameter	P10.1.4	B5:m	[]	Yes []
B5c	Calling Device ID parameter	P10.1.4	B5:m	[]	Yes []
B5d	Called Device parameter	P10.1.4	B5:m	[]	Yes []
B5e	Last Redirection Device parameter	P10.1.4	B5:m	[]	Yes []
B5f	Cause parameter	P10.1.4	B5:o		Yes [] No []
B5g	Local Connection information	P10.1.4	B5:o		Yes [] No []

A.7.1.6 Diverted

Item	Service / Feature	Reference	Status	N/A	Supported?
B6	Diverted event	S10.2.3.5 P10.1.5	o		Yes [] No []
B6a	Diverted Connection ID parameter	P10.1.5	B6:C.2	[]	Yes [] No []
B6b	Diverting Device parameter	P10.1.5	B6:m	[]	Yes []
B6c	New Destination parameter	P10.1.5	B6:m	[]	Yes []
B6d	Cause parameter	P10.1.5	B6:o		Yes [] No []
B6e	Local Connection information	P10.1.5	B6:o		Yes [] No []

C.2: If the call alerted the device then mandatory
else optional

A.7.1.7 Established

Item	Service / Feature	Reference	Status	N/A	Supported?
B7	Established event	S10.2.3.6 P10.1.6	o		Yes [] No []
B7a	Established Connection parameter	P10.1.6	B7:m	[]	Yes []
B7b	Answering Device parameter	P10.1.6	B7:m	[]	Yes []
B7c	Calling Device parameter	P10.1.6	B7:m	[]	Yes []
B7d	Called Device parameter	P10.1.6	B7:m	[]	Yes []
B7e	Last Redirection Device parameter	P10.1.6	B7:m	[]	Yes []
B7f	Cause parameter	P10.1.6	B7:o		Yes [] No []
B7g	Local Connection information	P10.1.6	B7:o		Yes [] No []

A.7.1.8 Failed

Item	Service / Feature	Reference	Status	N/A	Supported?
B8	Failed event	S10.2.3.7 P10.1.7	o		Yes [] No []
B8a	Failed Connection	P10.1.7	B8:m	[]	Yes []
B8b	Failing Device parameter	P10.1.7	B8:m	[]	Yes []
B8c	Called Device parameter	P10.1.7	B8:m	[]	Yes []
B8d	Cause parameter	P10.1.7	B8:o		Yes [] No []
B8e	Local Connection information	P10.1.7	B8:o		Yes [] No []

A.7.1.9 Held

Item	Service / Feature	Reference	Status	N/A	Supported?
B9	Held event	S10.2.3.8 P10.1.8	o		Yes [] No []
B9a	Held Connection	P10.1.8	B9:m	[]	Yes []
B9b	Holding Device parameter	P10.1.8	B9:m	[]	Yes []
B9c	Cause parameter	P10.1.8	B9:o		Yes [] No []
B9d	Local Connection information	P10.1.8	B9:o		Yes [] No []

A.7.1.10 Network Reached

Item	Service / Feature	Reference	Status	N/A	Supported?
B10	Network Reached event	S10.2.3.9 P10.1.9	o		Yes [] No []
B10a	Connection ID parameter	P10.1.9	B10:m	[]	Yes []
B10b	Trunk Used parameter	P10.1.9	B10:m	[]	Yes []
B10c	Called Device parameter	P10.1.9	B10:m	[]	Yes []
B10d	Cause parameter	P10.1.9	B10:o		Yes [] No []
B10e	Local Connection information	P10.1.9	B10:o		Yes [] No []

A.7.1.11 Originated

Item	Service / Feature	Reference	Status	N/A	Supported?
B11	Originated event	S10.2.3.10 P10.1.10	o		Yes [] No []
B11a	Originated Connection parameter	P10.1.10	B11:m	[]	Yes []
B11b	Calling Device parameter	P10.1.10	B11:m	[]	Yes []
B11c	Called Device parameter	P10.1.10	B11:m	[]	Yes []
B11d	Cause parameter	P10.1.10	B11:o		Yes [] No []
B11e	Local Connection information	P10.1.10	B11:o		Yes [] No []

A.7.1.12 Queued

Item	Service / Feature	Reference	Status	N/A	Supported?
B12	Queued event	S10.2.3.11 P10.1.11	o		Yes [] No []
B12a	Queued Connection parameter	P10.1.11	B12:m	[]	Yes []
B12b	Queue parameter	P10.1.11	B12:m	[]	Yes []
B12c	Calling Device parameter	P10.1.11	B12:m	[]	Yes []
B12d	Called Device parameter	P10.1.11	B12:m	[]	Yes []
B12e	Last Redirection Device parameter	P10.1.11	B12:m	[]	Yes []
B12f	Number of Calls in Queue	P10.1.11	B12:o		Yes [] No []
B12g	Cause parameter	P10.1.11	B12:o		Yes [] No []
B12h	Local Connection information	P10.1.11	B12:o		Yes [] No []

A.7.1.13 Retrieved

Item	Service / Feature	Reference	Status	N/A	Supported?
B13	Retrieved event	S10.2.3.12 P10.1.12	o		Yes [] No []
B13a	Retrieved Connection parameter	P10.1.12	B13:m	[]	Yes []
B13b	Retrieving Device parameter	P10.1.12	B13:m	[]	Yes []
B13c	Cause parameter	P10.1.12	B13:o		Yes [] No []
B13d	Local Connection information	P10.1.12	B13:o		Yes [] No []

A.7.1.14 Service Initiated

Item	Service / Feature	Reference	Status	N/A	Supported?
B14	Service Initiated event	S10.2.3.13 P10.1.13	o		Yes [] No []
B14a	Initiated Connection parameter	P10.1.13	B14:m	[]	Yes []
B14b	Cause parameter	P10.1.13	B14:o		Yes [] No []
B14c	Local Connection information	P10.1.13	B14:o		Yes [] No []

A.7.1.15 Transferred

Item	Service / Feature	Reference	Status	N/A	Supported?
B15	Transferred event	S10.2.3.14 P10.1.14	o		Yes [] No []
B15a	Primary Old Call parameter	P10.1.14	B15:m	[]	Yes []
B15b	Secondary Old Call parameter	P10.1.14	B15:C.3	[]	Yes [] No []
B15c	Transferring Device parameter	P10.1.14	B15:m	[]	Yes []
B15d	Transferred-to Device parameter	P10.1.14	B15:m	[]	Yes []
B15e	Cause parameter	P10.1.14	B15:o		Yes [] No []
B15f	Local Connection information	P10.1.14	B15:o		Yes [] No []
B15g	Connection ID list	P10.1.14	B15:o		Yes [] No []
B15h	Static Device ID included in list	P16.4	B15g:o	[]	Yes [] No []

C.3: If parameter in previous events then mandatory
else optional

A.7.2 Agent Events

A.7.2.1 Logged On

Item	Service / Feature	Reference	Status	N/A	Supported?
B16	Logged On event	S10.2.2.1 P10.3.1	o		Yes [] No []
B16a	Agent Device parameter	P10.3.1	B16:m	[]	Yes []
B16b	Agent ID parameter	P10.3.1	B16:o		Yes [] No []
B16c	Agent Group parameter	P10.3.1	B16:o		Yes [] No []
B16d	Password parameter	P10.3.1	B16:o		Yes [] No []

A.7.2.2 Logged Off

Item	Service / Feature	Reference	Status	N/A	Supported?
B17	Logged Off event	S10.2.2.2 P10.3.2	o		Yes [] No []
B17a	Agent Device parameter	P10.3.2	B17:m	[]	Yes []
B17b	Agent ID parameter	P10.3.2	B17:o		Yes [] No []
B17c	Agent Group parameter	P10.3.2	B17:o		Yes [] No []

A.7.2.3 Ready

Item	Service / Feature	Reference	Status	N/A	Supported?
B18	Ready event	S10.2.2.4 P10.3.4	o		Yes [] No []
B18a	Agent Device parameter	P10.3.4	B18:m	[]	Yes []
B18b	Agent ID parameter	P10.3.4	B18:o		Yes [] No []

A.7.2.4 Not Ready

Item	Service / Feature	Reference	Status	N/A	Supported?
B19	Not Ready event	S10.2.2.3 P10.3.3	o		Yes [] No []
B19a	Agent Device parameter	P10.3.3	B19:m	[]	Yes []
B19b	Agent ID parameter	P10.3.3	B19:o		Yes [] No []

A.7.2.5 Work Ready

Item	Service / Feature	Reference	Status	N/A	Supported?
B20	Work Ready event	P10.2.2.6 P10.3.6	o		Yes [] No []
B20a	Agent Device parameter	P10.3.6	B20:m	[]	Yes []
B20b	Agent ID parameter	P10.3.6	B20:o		Yes [] No []

A.7.2.6 Work Not Ready

Item	Service / Feature	Reference	Status	N/A	Supported?
B21	Work Not Ready event	P10.2.2.5 P10.3.5	o		Yes [] No []
B21a	Agent Device parameter	P10.3.5	B21:m	[]	Yes []
B21b	Agent ID parameter	P10.3.5	B21:o		Yes [] No []

A.7.3 Other Feature Events

A.7.3.1 Call Information

Item	Service / Feature	Reference	Status	N/A	Supported?
B22	Call Information event	S10.2.4.1 P10.2.1	o		Yes [] No []
B22a	Connection ID parameter	P10.2.1	B22:m	[]	Yes []
B22b	Device parameter	P10.2.1	B22:o		Yes [] No []
B22c	Account Information	P10.2.1	B22:o		Yes [] No []
B22d	Authorisation Code	P10.2.1	B22:o		Yes [] No []

A.7.3.2 Do Not Disturb

Item	Service / Feature	Reference	Status	N/A	Supported?
B23	Do Not Disturb event	S10.2.4.2 P10.2.2	o		Yes [] No []
B23a	Device parameter	P10.2.2	B23:m	[]	Yes []
B23b	Do Not Disturb On	P10.2.2	B23:m	[]	Yes []

A.7.3.3 Forwarding

Item	Service / Feature	Reference	Status	N/A	Supported?
B24	Forwarding event	S10.2.4.3 P10.2.3	o		Yes [] No []
B24a	Device parameter	P10.2.3	B24:m	[]	Yes []
B24b	Forwarding Information	P10.2.3	B24:m	[]	Yes []
B24c	Forwarding Type parameter	P16.6	B24b:m	[]	Yes []
B24d	Forward Immediate On	P16.6	B24c:o	[]	Yes [] No []
B24e	Forward Immediate Off	P16.6	B24c:o	[]	Yes [] No []
B24f	Forward Busy On	P16.6	B24c:o	[]	Yes [] No []
B24g	Forward Busy Off	P16.6	B24c:o	[]	Yes [] No []
B24h	Forward No Answer On	P16.6	B24c:o	[]	Yes [] No []
B24i	Forward No Answer Off	P16.6	B24c:o	[]	Yes [] No []
B24j	Forward Busy Internal On	P16.6	B24c:o	[]	Yes [] No []
B24k	Forward Busy Internal Off	P16.6	B24c:o	[]	Yes [] No []
B24l	Forward Busy External On	P16.6	B24c:o	[]	Yes [] No []
B24m	Forward Busy External Off	P16.6	B24c:o	[]	Yes [] No []
B24n	Fwd No Answer Internal On	P16.6	B24c:o	[]	Yes [] No []
B24o	Fwd No Answer Internal Off	P16.6	B24c:o	[]	Yes [] No []
B24p	Fwd No Answer External On	P16.6	B24c:o	[]	Yes [] No []
B24q	Fwd No Answer External Off	P16.6	B24c:o	[]	Yes [] No []
B24r	Forward DN	P16.6	B24b:o		Yes [] No []

A.7.3.4 Message Waiting

Item	Service / Feature	Reference	Status	N/A	Supported?
B25	Message Waiting event	S10.2.4.4 P10.2.4	o		Yes [] No []
B25a	Device for Message	P10.2.4	B25:m	[]	Yes []
B25b	Invoking Device	P10.2.4	B25:m	[]	Yes []
B25c	Message Waiting On parameter	P10.2.4	B25:m	[]	Yes []

A.7.4 Maintenance Events

A.7.4.1 Back In Service

Item	Service / Feature	Reference	Status	N/A	Supported?
B26	Back in Service event	S10.2.5.1 P10.4.1	o		Yes [] No []
B26a	Device ID parameter	P10.4.1	B26:m	[]	Yes []
B26b	Cause parameter	P10.4.1	B26:o		Yes [] No []

A.7.4.2 Out of Service

Item	Service / Feature	Reference	Status	N/A	Supported?
B27	Out of Service event	S10.2.5.2 P10.4.2	o		Yes [] No []
B27a	Device ID parameter	P10.4.2	B27:m	[]	Yes []
B27b	Cause parameter	P10.4.2	B27:o		Yes [] No []

A.7.5 Private Events

A.7.5.1 Private

Item	Service / Feature	Reference	Status	N/A	Supported?
B28	Is/Are Private Event(s)	S10.2.1 P10.5	o		Yes [] No []

A.8 Computing Services

A.8.1 Route Request

Item	Service / Feature	Reference	Status	N/A	Supported?
C1	Route Request service	S11.3 P11.1	o		Yes [] No []
C1a	Cross Reference parameter	P11.1	C1:m	[]	Yes []
C1b	Current Route parameter	P11.1	C1:m	[]	Yes []
C1c	Calling Device parameter	P11.1	C1:o		Yes [] No []
C1d	Routed Call parameter	P11.1	C1:o		Yes [] No []
C1e	Route Select Algorithm	P11.1	C1:o		Yes [] No []
C1f	Priority parameter	P11.1	C1:o		Yes [] No []
C1g	Set-up information	P11.1	C1:o		Yes [] No []
C1h	Security parameters	P16.8	C1:o		Yes [] No []
C1i	Private Data in Request	P16.8	C1:o		Yes [] No []
C1j	Report of Service Errors	P14	C1:m	[]	Yes []

A.8.2 Re-Route Service

Item	Service / Feature	Reference	Status	N/A	Supported?
C2	Re-Route service	S11.1 P11.2	o		Yes [] No []
C2a	Cross Reference parameter	P11.2	C2:m	[]	Yes []
C2b	Security parameters	P16.8	C2:o		Yes [] No []
C2c	Private Data in Request	P16.8	C2:o		Yes [] No []
C2d	Report of Service Errors	P14	C2:m	[]	Yes []

A.8.3 Route Select

Item	Service / Feature	Reference	Status	N/A	Supported?
C3	Route Select service	S11.4 P11.3	o		Yes [] No []
C3a	Cross Reference parameter	P11.3	C3:m	[]	Yes []
C3b	Route Selected parameter	P11.3	C3:m	[]	Yes []
C3c	ISDN Set-Up parameter	P11.3	C3:o		Yes [] No []
C3d	Remaining Retries parameter	P11.3	C3:o		Yes [] No []
C3e	Route Used Request parameter	P11.3	C3:o		Yes [] No []
C3f	Security parameters	P16.8	C3:o		Yes [] No []
C3g	Private Data in Request	P16.8	C3:o		Yes [] No []
C3h	Report of Service Errors	P14	C3:m	[]	Yes []

A.8.4 Route Used

Item	Service / Feature	Reference	Status	N/A	Supported?
C4	Route Used service	S11.5 P11.4	o		Yes [] No []
C4a	Cross Reference parameter	P11.4	C4:m	[]	Yes []
C4b	Route Used parameter	P11.4	C4:m	[]	Yes []
C4c	Calling Device parameter	P11.4	C4:o		Yes [] No []
C4d	Domain parameter	P11.4	C4:o		Yes [] No []
C4e	Security parameters	P16.8	C4:o		Yes [] No []
C4f	Private Data in Request	P16.8	C4:o		Yes [] No []
C4g	Report of Service Errors	P14	C4:m	[]	Yes []

A.8.5 Route End

Item	Service / Feature	Reference	Status	N/A	Supported?
C5	Route End service	S11.2 P11.5	o		Yes [] No []
C5a	Cross Reference parameter	P11.5	C5:m	[]	Yes []
C5b	Error Value parameter	P11.5	C5:o		Yes [] No []
C5c	Security parameters	P16.8	C5:o		Yes [] No []
C5d	Private Data in Request	P16.8	C5:o		Yes [] No []
C5e	Report of Service Errors	P14	C5:m	[]	Yes []

A.9 Bidirectional Services

A.9.1 Escape Service

Item	Service / Feature	Reference	Status	N/A	Supported?
D1	Escape service	S12.1 P12.1	o		Yes [] No []
D1a	Service Result	P5.3 P12.1	D1:m	[]	Yes []
D1b	Security parameters	P16.8	D1:o		Yes [] No []
D1c	Private Data in Request	P16.8	D1:o		Yes [] No []
D1d	Private Data in Result	P16.8	D1:o		Yes [] No []
D1e	Report of Service Errors	P14	D1:m	[]	Yes []

A.9.2 System Status

Item	Service / Feature	Reference	Status	N/A	Supported?
D2	System Status service	S12.2 P12.2	o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2a	Service Result	P5.3 P12.2	D2:m	<input type="checkbox"/>	Yes <input type="checkbox"/>
D2b	System Status Cause	P12.2	D2:m	<input type="checkbox"/>	Yes <input type="checkbox"/>
D2c	Initializing	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2d	Enabled	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2e	Normal	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2f	Messages Lost	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2g	Disabled	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2h	Overload Imminent	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2i	Overload Reached	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2j	Overload Relieved	P12.2	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2k	Security parameters	P16.8	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2l	Private Data in Request	P16.8	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2m	Private Data in Result	P16.8	D2:o		Yes <input type="checkbox"/> No <input type="checkbox"/>
D2n	Report of Service Errors	P14	D2:m	<input type="checkbox"/>	Yes <input type="checkbox"/>

A.10 Status Reporting Services

A.10.1 Change Monitor Filter

Item	Service / Feature	Reference	Status	N/A	Supported?
E1	Change Monitor Filter service	S10.1 P13.2	o		Yes [] No []
E1a	Service Result	P5.3 P13.2	E1:m	[]	Yes []
E1b	Cross Reference ID parameter in Request	P13.2	E1:m	[]	Yes []
E1c	Filter List in Request	P13.2	E1:m	[]	Yes []
E1d	Filter List in Result	P13.2	E1:o		Yes [] No []
E1e	Security parameters	P16.8	E1:o		Yes [] No []
E1f	Private Data in Request	P16.8	E1:o		Yes [] No []
E1g	Private Data in Result	P16.8	E1:o		Yes [] No []
E1h	Report of Service Errors	P14	E1:m	[]	Yes []

A.10.2 Monitor Start

Item	Service / Feature	Reference	Status	N/A	Supported?
E2	Monitor Start	S10.3 P13.1	o		Yes [] No []
E2a	Service Result	P5.3 P13.1	E2:m	[]	Yes []
E2b	Monitor Object parameter	P13.1	E2:m	[]	Yes []
E2c	Monitor Object device	P13.1	E2:o1		Yes [] No []
E2d	Monitor Object call	P13.1	E2:o1		Yes [] No []
E2e	Monitor Type parameter	P13.1	E2:o		Yes [] No []
E2f	Monitor Type device	P13.1	c1:o1		Yes [] No []
E2g	Monitor Type call	P13.1	c1:o1		Yes [] No []
E2h	Monitor Type default to Call Monitoring	P13.1	E2:o1		Yes [] No []
E2i	Monitor Type default to Device Monitoring	P13.1	E2:o1		Yes [] No []
E2j	Monitor Filter parameter for Call Processing events	P13.1	E2:o		Yes [] No []
E2k	Monitor Filter parameter for Feature events	P13.1	E2:o		Yes [] No []
E2l	Monitor Filter parameter for Agent events	P13.1	E2:o		Yes [] No []
E2m	Monitor Filter parameter for Maintenance events	P13.1	E2:o		Yes [] No []
E2n	Cross Reference ID in Service Result	P13.1	E2a:m	[]	Yes []
E2o	Security parameters	P16.8	E2:o		Yes [] No []
E2p	Private Data in Request	P16.8	E2:o		Yes [] No []
E2q	Private Data in Result	P16.8	E2:o		Yes [] No []
E2r	Report of Service Errors	P14	E2:m	[]	Yes []

Note A.4

c1: (E2c or E2d)

A.10.3 Monitor Stop

Item	Service / Feature	Reference	Status	N/A	Supported?
E3	Monitor Stop	S10.4 P13.3	o		Yes [] No []
E3a	Service Result	P5.3 P13.3	E3:m	[]	Yes []
E3b	Cross Reference ID parameter in Request	P13.3	E3:m	[]	Yes []
E3c	Security parameters	P16.8	E3:o		Yes [] No []
E3d	Private Data in Request	P16.8	E3:o		Yes [] No []
E3e	Private Data in Result	P16.8	E3:o		Yes [] No []
E3f	Report of Service Errors	P14	E3:m	[]	Yes []

A.10.4 Snapshot Call

Item	Service / Feature	Reference	Status	N/A	Supported?
E4	Snapshot Call service	S10.5 P13.5	o		Yes [] No []
E4a	Service Result	P5.3 P13.5	E4:m	[]	Yes []
E4b	Snapshot Object parameter in Request	P13.5	E4:m	[]	Yes []
E4c	Static Device ID parameter in Result	P13.5	E4a:m	[]	Yes []
E4d	Connection ID parameter in Result	P13.5	E4a:m	[]	Yes []
E4e	Local Connection State parameter in Result	P13.5	E4a:o		Yes [] No []
E4f	Security parameters	P16.8	E4:o		Yes [] No []
E4g	Private Data in Request	P16.8	E4:o		Yes [] No []
E4h	Private Data in Result	P16.8	E4:o		Yes [] No []
E4i	Report of Service Errors	P14	E4:m	[]	Yes []

A.10.5 Snapshot Device

Item	Service / Feature	Reference	Status	N/A	Supported?
E5	Snapshot Device service	S10.6 P13.4	o		Yes [] No []
E5a	Service Result	P5.3 P13.4	E5:m	[]	Yes []
E5b	Static Device ID parameter in Request	P13.4	E5a:m	[]	Yes []
E5c	Connection ID parameter in Result	P13.4	E5a:m	[]	Yes []
E5d	Call State parameter in Result	P13.4	E5a:m	[]	Yes []
E5e	Security parameters	P16.8	E5:o		Yes [] No []
E5f	Private Data in Request	P16.8	E5:o		Yes [] No []
E5g	Private Data in Result	P16.8	E5:o		Yes [] No []
E5h	Report of Service Errors	P14	E5:m	[]	Yes []

A.11 Switching Event Cause Values

Item	Cause	Reference	Status	N/A	Supported?
F1	Cause values in event reports	S10.2.8 P15	o		Yes [] No []
F1a	Active Monitor	P15	F1:o		Yes [] No []
F1b	Alternate	P15	F1:o		Yes [] No []
F1c	Busy	P15	F1:o		Yes [] No []
F1d	Call Back	P15	F1:o		Yes [] No []
F1e	Call Cancelled	P15	F1:o		Yes [] No []
F1f	Call Forward Immediate	P15	F1:o		Yes [] No []
F1g	Call Forward Busy	P15	F1:o		Yes [] No []
F1h	Call Forward No Answer	P15	F1:o		Yes [] No []
F1i	Call Forward	P15	F1:o		Yes [] No []
F1j	Call Not Answered	P15	F1:o		Yes [] No []
F1k	Call Pickup	P15	F1:o		Yes [] No []
F1l	Camp On	P15	F1:o		Yes [] No []
F1m	Destination Not Obtainable	P15	F1:o		Yes [] No []
F1n	Do Not Disturb	P15	F1:o		Yes [] No []
F1o	Incompatible Destination	P15	F1:o		Yes [] No []
F1p	Invalid Account Code	P15	F1:o		Yes [] No []
F1q	Key Operation	P15	F1:o		Yes [] No []
F1r	Lockout	P15	F1:o		Yes [] No []
F1s	Maintenance	P15	F1:o		Yes [] No []
F1t	Network Congestion	P15	F1:o		Yes [] No []
F1u	Network Not Obtainable	P15	F1:o		Yes [] No []
F1v	New Call	P15	F1:o		Yes [] No []
F1w	No Available Agents	P15	F1:o		Yes [] No []
F1x	Overflow	P15	F1:o		Yes [] No []

A.11 Switching Event Cause Values (continued)

Item	Cause	Reference	Status	N/A	Supported?
F1y	Override	P15	F1:o		Yes [] No []
F1z	Park	P15	F1:o		Yes [] No []
F1aa	Recall	P15	F1:o		Yes [] No []
F1ab	Redirected	P15	F1:o		Yes [] No []
F1ac	Reorder Tone	P15	F1:o		Yes [] No []
F1ad	Resources Not Available	P15	F1:o		Yes [] No []
F1ae	Silent Monitor	P15	F1:o		Yes [] No []
F1af	Transfer	P15	F1:o		Yes [] No []
F1ag	Trunks Busy	P15	F1:o		Yes [] No []
F1ah	Voice Unit Initiator	P15	F1:o		Yes [] No []

A.12 Switching Function Errors

A.12.1 Operational Errors

Item	Error	Reference	Status	N/A	Supported?
G1	Operational Errors category	S8.4.1 P14	o		Yes [] No []
G1a	Generic	P14	G1:o		Yes [] No []
G1b	Request Incompatible with Object	P14	G1:o		Yes [] No []
G1c	Value Out of Range	P14	G1:o		Yes [] No []
G1d	Object Not Known	P14	G1:o		Yes [] No []
G1e	Invalid Calling Device	P14	G1:o		Yes [] No []
G1f	Invalid Called Device	P14	G1:o		Yes [] No []
G1g	Privilege Violation on Specified Device	P14	G1:o		Yes [] No []
G1h	Invalid Forwarding Destination	P14	G1:o		Yes [] No []
G1i	Privilege Violation on Called Device	P14	G1:o		Yes [] No []
G1j	Privilege Violation on Calling Device	P14	G1:o		Yes [] No []
G1k	Invalid Call Identifier	P14	G1:o		Yes [] No []
G1l	Invalid Device Identifier	P14	G1:o		Yes [] No []
G1m	Invalid Connection Identifier	P14	G1:o		Yes [] No []
G1n	Invalid Destination	P14	G1:o		Yes [] No []
G1o	Invalid Feature	P14	G1:o		Yes [] No []
G1p	Invalid Allocation State	P14	G1:o		Yes [] No []
G1q	Invalid Cross Reference ID	P14	G1:o		Yes [] No []
G1r	Invalid Object Type	P14	G1:o		Yes [] No []
G1s	Security Violation	P14	G1:o		Yes [] No []

A.12.2 State Incompatibility

Item	Error	Reference	Status	N/A	Supported?
G2	State Incompatibility category	S8.4.3 P14	o		Yes [] No []
G2a	Generic	P14	G2:o		Yes [] No []
G2b	Incorrect Object State	P14	G2:o		Yes [] No []
G2c	Invalid Connection ID	P14	G2:o		Yes [] No []
G2d	No Active Call	P14	G2:o		Yes [] No []
G2e	No Held Call	P14	G2:o		Yes [] No []
G2f	No Call to Clear	P14	G2:o		Yes [] No []
G2g	No Connection to Clear	P14	G2:o		Yes [] No []
G2h	No Call to Answer	P14	G2:o		Yes [] No []
G2i	No Call to Complete	P14	G2:o		Yes [] No []

A.12.3 System Resource Availability Errors

Item	Error	Reference	Status	N/A	Supported?
G3	System Resource Availability Errors category	S8.4.4 P14	o		Yes [] No []
G3a	Generic	P14	G3:o		Yes [] No []
G3b	Service Busy	P14	G3:o		Yes [] No []
G3c	Resource Busy	P14	G3:o		Yes [] No []
G3d	Resource Out of Service	P14	G3:o		Yes [] No []
G3e	Network Busy	P14	G3:o		Yes [] No []
G3f	Network Out of Service	P14	G3:o		Yes [] No []
G3g	Overall Monitor Limit Exceeded	P14	G3:o		Yes [] No []
G3h	Conference Member Limit Exceeded	P14	G3:o		Yes [] No []

A.12.4 Subscribed Resource Availability Errors

Item	Error	Reference	Status	N/A	Supported?
G4	Subscribed Resource Availability Errors category	S8.4.5 P14	o		Yes [] No []
G4a	Generic	P14	G4:o		Yes [] No []
G4b	Object Monitor Limit Exceeded	P14	G4:o		Yes [] No []
G4c	External Trunk Limit Exceeded	P14	G4:o		Yes [] No []
G4d	Outstanding Request Limit Exceeded	P14	G4:o		Yes [] No []

A.12.5 Performance Errors

Item	Error	Reference	Status	N/A	Supported?
G5	Performance Errors category	S8.4.6 P14	o		Yes [] No []
G5a	Generic	P14	G5:o		Yes [] No []
G5b	Performance Limit Exceeded	P14	G5:o		Yes [] No []

A.13 CSTA Data Types

Item	Data Type	Reference	Status	N/A	Supported?
H1	Dynamic Device IDs	S6.1.1 P16.3	o		Yes [] No []
H2	Extended Device IDs (choices follow)	S6.1.1 P16.2	m	[]	Yes []
H2a	Device Identifier	P16.2	H2:o1		Yes [] No []
H2b	Implicit Public	P16.2	H2:o1		Yes [] No []
H2c	Explicit Public	P16.2	H2:o1		Yes [] No []
H2d	Implicit Private	P16.2	H2:o1		Yes [] No []
H2e	Explicit Private	P16.2	H2:o1		Yes [] No []
H2f	Other plan	P16.2	H2:o1		Yes [] No []
H3	Device IDs (choices follow)	S6.1.1 P16.2	m	[]	Yes []
H3a	Number digits	P16.2	H3:o1		Yes [] No []
H3b	Device number	P16.2	H3:o1		Yes [] No []

A.14 Security

Item	Service / Feature	Reference	Status	N/A	Supported?
I1	Security option	S7 P7 P16.7	o		Yes [] No []
I1a	Message Sequence Number	P16.7	I1:o		Yes [] No []
I1b	Time Stamp	P16.7	I1:o		Yes [] No []
I1c	Privilege Attribute Certificate (PAC)	P16.7	I1:o		Yes [] No []
I1d	Seal	P16.7	I1:o		Yes [] No []

This Standard ECMA-180 is available free of charge from:

**ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland**

**Fax: +41 22 849.60.01
Internet: helpdesk@ecma.ch**

This Standard can also be downloaded as files E180-DOC.EXE or E180-PSC.EXE from ECMANEWS