CSTA V2.7.1

Private information – ASN.1 encoding

Order number	3BA 19598 CIAA
OmniPCX Enterprise release	6.x / 7.x
Revised	January 2008

Notice

While reasonable effort was made to ensure that the information in this document was complete and accurate at the time of printing, we can not assume responsibility for any errors. Changes and/or corrections to the information contained in this document may be incorporated into future issues.

Information

Alcatel-Lucent, Technical Support and Services
1, rue du Docteur Albert Schweitzer
67408 Illkirch – France

http://www.applicationpartner.alcatel-lucent.com/

Contents

Supprimé : 75
Supprimé : 76
Supprimé : 75
Supprimé : 77
Supprimé : 75
Supprimé : 78
Supprimé : 75
Supprimé : 78
Supprimé : 75
Supprimé : 79
Supprimé : 75
Supprimé : 80
Supprimé : 75
Supprimé : 81
Supprimé : 75
Supprimé : 82
Supprimé : 75
Supprimé : 83
Supprimé : 75
Supprimé : 84
Supprimé : 75
Supprimé : 85
Supprimé : 75
Supprimé : 87
Supprimé : 75
Supprimé : 93

Chapter 1

Introduction ASN.1

1.1 Objective

The purpose of the Alcatel-Lucent CSTA private information is to anticipate new CSTA standard phases and provide an Alcatel-Lucent specific added value. Each information item must have its own object identifier and is intended to be decoded by any computing platform's switch driver to be included in the API data structures. The "other" type information is reserved for very specific situations like:

- conveying application specific data which do not have any generic utility,
- quickly implementing (or testing) a new feature by tunneling information between the switch and the application.

This information must be used very carefully as the data structure is under the responsibility of the end points. The issues are:

- compatibility,
- data organization (little/big Indian),
- more complex processing for the application.

This document lists the Alcatel-Lucent private information used in the various CSTA versions. However, complete information is not available in all switch versions. For example, in release R1.4 of the OmniPCX 4400 switch, the *Associate Data* private service exists but is defined in the CSTA version 2 protocol. This private service is therefore no longer necessary in a 2.0 release running CSTA version 2.

Section 2.4 summarizes the Alcatel-Lucent private information and lists the CSTA version where they are implemented.

Section 2.5 summarizes the private data used in the CSTA events and in the CSTA services.

Related documents

[1] Standard ECMA – 217 Services for Computer Supported Telecommunications Applications (CSTA) Phase II (December 1994)

[2] Standard ECMA – 218 Protocol for Computer Supported Telecommunications Applications (CSTA) Phase II (December 1994)

1.2 What's new

Version 2.7.1 of CSTA introduces the following new private services and private data:

- Call Center Treatment Request
- Start Data Collection
- Stop Data Collection
- Data Collected
- Generate Telephony Tones
- Cancel Telephony Tones
- Reroute Authorization
- Additional Digits Reporting Criteria
- Announcements or Music

The new generated private events introduced in this release include:

- Generated Telephony Tones

1.3 Alcatel-Lucent object identifiers

Alcatel-Lucent private information is organized with the following object identifiers: 1.3.0012.1206.xxx where xxx can take the following values:

- ***operations (0 - 499) - the operation name is followed by the ASN1 encoding name:***
  - 0 : Associate Data - associateData (not used: defined in ECMA V2)
  - 1 : Set Date And Time - dateAndTimeSetting (not used from OmniPCX Enterprise release R4.1)
  - 2 : Host Information - hostInformation (old ASN1 name: IVRformation)
  - 3 : Interrupt Transaction - interruptTransaction
  - 4 : Set Device In Service - setDevicel n Service
  - 5 : Fast Data - fastData (not used: defined in ECMA V2)
  - 6 : Supervisor Assist Request - supervisorAssistRequest
  - 7 : Supervisor Assist Cancel - supervisorAssistCancel
  - 8 : Escape Register Request - serviceRegisterRequest
  - 9 : Escape Register Cancel - serviceRegisterCancel
  - 10 : Agent Logging Information - agentLoggingInformation
  - 11 : Reserved
  - 12 : Reserved
  - 13 : Reserved
  - 14 : Roaming DECT Status - roamingDectStatus
  - 15 : Permanent Listening Activation - permanentListeningActivation
  - 16 : Service Escape Call Center Treatment Request serviceEscapeCCTreatmentRequest
  - 17 : Set Message Waiting Indicator - setMessageWaitingIndicator
  - 18 : Service Escape Start Data Collection - serviceEscapeStartDataCollection

- 19 : Service Escape Stop Data Collection - serviceEscapeStopDataCollection
- 20 : Service Escape Data Collected - serviceEscapeDataCollected
- 21 : Service Escape Generate Telephony Tones - serviceEscapeGenerateTelephonyTones
- 22 : Service Escape Cancel Telephony Tones - serviceEscapeCancelTelephonyTones
- 23-498 : reserved
- 499 : Other Operation - otherOperation

- ***private data (500 - 999) - the data name is followed by the ASN1 encoding name:***
  - 500 : Correlator Data - correlatorData (not used: defined in ECMA V2)
  - 501 : Private Errors - privateErrors (not used: defined in ECMA V2)
  - 502 : CCD Pilot Expected Waiting Time - acdWaitingTime
  - 503 : Service Option - cstaServiceOption
  - 504 : Network Time Slot - networkTimeSlot
  - 505 : Not Ready Context - notReadyContext
  - 506 : CCD Agent Assignment Information - queryAgentInformation
  - 507 : CCD Pilot Status - pilotTransferInfo
  - 508 : CCD Delivered Call Information - acdDistributionInfo
  - 509 : Interactive Queueing Delivered Call Information - iVRInteractiveQueueing
  - 510 : Interactive Queueing Reconnect Guide Level - iVRNextLevel
  - 511 : Withdrawal Type - notReadyActivation
  - 512 : Party Name - partyName
  - 513 : Requesting Device - requestingDevice
  - 514 : Rerouted Call Indication - acdNetworkRerouted
  - 515 : IVR Host Identification - hostIvrIdentification
  - 516 : Supervised Transfer - supervisedTransfer
  - 517 : reserved
  - 518 : reservedACR Attribute list - acrAttributeList
  - 519 : CCD Treatment Type- cCTreatmentType
  - 520 : Secrecy Identity - secretIdentiteList
  - 521 : Universal Call Identifier - globalCallID
  - 522 : Universal Call Identifier Reminder - oldGlobalCallID
  - 523 : National/International Indicator- nationalIndication

- 524 : Reroute Authorization - rerouteAuthorisation
- 525 : Additional Digits Reporting Criteria - digitsReportingCriteriaAdd
- 526 : Announcement or Music - announcementOrMusic
- 527 : Global Call ID List - globalCallIDList
- 528 : localHybridLinkFlag
- 529 : agent type for loggon
- 530 : pilot number or rsi number
- 531-998 : reserved
- 999 : Other Private Data - otherPrivateData

- ***private events (1000 - 1499) - the event name is followed by the ASN1 encoding name:***
  - 1000 : Head Of Queue - headOfQueueEvent
  - 1001 : Supervisor Assist Request - supervisorAssistRequestEvent
  - 1002 : Reserved
  - 1003 : Roaming DECT - roamingDectEvent
  - 1004 : Telephony Tones Generated Event - telephonyTonesGeneratedEvent
  - 1005 : Busy Event - busyEvent
  - 1006 : Remote Record Failed Event - remoteRecordFailedEvent
  - 1007-1498 : reserved
  - 1499 : Other Private Event - otherPrivateEvent

Any new CSTA private information will be included in this range and the corresponding ASN.1 description added to this document.

According to the ECMA standard, the reserved values as well as any non supported private information will be skipped by the ASN.1 decoder if received.

1.4 Private identifiers

Each set of "other" type information (i.e. *otherOperation, otherPrivateData* and *otherPrivateEven4* is identified *by a privateidentifier* field. This private identifier relates to an application type. It is underthe application's responsibility to define sub-types. The purpose of a private identifier is also to allow byte ordering detection. These values are centralized in this document..

At the moment, the following values for private identifiers are defined:

```
#define   PRIV ACD_DATA      OxCACD   /*Identifier of CCD private information*/
#define   PRIV ACD_IVR       OxCACE   /*Identifier of CCD private information*/
#define   PRIV_PC_DATA       OxCACF   /*Identifier of A4980 private information*/
#define   PRIV_VIOLA DATA    OxCASA   /*Identifier of CCivr private information*/
#define   PRIV_LIB_BUSY      OxCA1 C  /*CSTA Divert Call service specific data*/
```

Note:

The private identifier PRIV_LIB_BUSY is used to reject an external TO/T2 call to the public network with a "busy" CSTA cause. This feature may be used by an external routing application based on the CSTA Divert Call service; for example:

- to clear the external call when all agents of a Call Centre are busy,
- to re-route the external call in the network.

Chapter 2

Private Information

The following section defines all the types needed for private operations, private data and private events:

CSTA-extend
{ extension-types (129) }
DEFINITIONS::=
BEGIN
EXPORTS
CSTACommonArguments, CSTAPrivateData, Alcatel-LucentPrivateEventInfo;

IMPORTS

CSTAObject FROM CSTA-switch
{ iso identified-organization icd-ecma standard csta2 switching-function-objects }

ConnectionID FROM CSTA-call
{ iso identified-organization icd-ecma standard csta2 call-connection-identifiers }

SubjectDeviceID, DeviceID FROM CSTA-device
{ iso identified-organization icd-ecma standard csta2 device-identifiers }

AccountInfo, AuthCode, AgentGroup, CorrelatorData FROM CSTA-feature
{ iso identified-organization icd-ecma standard csta2 device -feature -types}

UniversalFailure, universalFailure FROM CSTA-error
{ iso identified-organization icd-ecma standard csta2 error-definition }

CSTASecurityData FROM *CSTA*-secure
{ iso identified-organization icd-ecma standard csta2 security };

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

AlcateItPrivateEventInfo::= SEQUENCE
{
   manufacturer          OBJECT IDENTIFIER, ANY DEFINED BY manufacturer
}

CSTAPrivateData::= CHOICE
{
   --The actuel encoding of the private event is added here,
   --replacing NULL with another valid ASN.1 type.
   privateData AlcatelCSTAPrivateData
}

AlcatelCSTAPrivateData ::= SEQUENCE
{
   manufacturer          OBJECT IDENTIFIER,
   ANY DEFINED BY manufacturer
}
```

From release R5.0 Ux, additional types (defined in ECMA3) are also required for RSI feature support:

```
ConnectionID3 ::= [APPLICATION11] CHOICE
{
callID[0]                       IMPLICITCallID3,
deviceID                        [1]  LocalDeviceID3,
both SEQUENCE
{  called                       [0]  IMPLICITCallID3,
   deviceID                     [1]LocalDeviceID3 }
{
CallID3 ::= OCTET STRING (SIZE(0..8))
LocalDeviceID3 ::=CHOICE
{  staticID                     Device                          ID3
   dynamicID                    [3]  IMPLICIT OCTET STRING (SIZE(0..32)) }

DeviceID3 ::= SEQUENCE
{
deviceIdentifier CHOICE
{  dialingNumber                [0]  IMPLICIT NumberDigits3,
   deviceNumber                 [1]  IMPLICIT DeviceNumber3,
   implicitPublic               [2]  IMPLICIT NumberDigits3,
   explicitPublic               [3]  PublicTON3,
   implicitPrivate              [4]  IMPLICIT NumberDigits3,
   explicitPrivate              [5]  PrivateTON3,
   other                        [6]  IMPLICIT OtherPlan3 },
   mediaClass                   MediaClass3 OPTIONAL }
PublicTON3 ::= 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 CCITTE E.164 --
```

```
PrivateTON3 ::= 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—

OtherPlan3 ::= OCTET STRING

--Allows future expansion to cover other numbering plans such as X.121

NumberDigits3 ::= IA5String

DeviceNumber3 ::= INTEGER

CSTASecurityData3 ::= SEQUENCE
{   messageSequenceNumber  [0]  IMPLICIT INTEGER   OPTIONAL
    timestamp                   TimInfo3           OPTIONAL
    securityInfo                SecurityInfo3      OPTIONAL }

SecurityInfo3 ::= CHOICE
{   string                  OCTET STRING
    privat                  NULL }
--The actual encoding is added here,
-- replacing NULL with another valid ASN.1 type

TimeInfo3 ::= UTCTime

ConnectionInformation3 ::= SEQUENCE
{flowDirection              ::= ENUMERATED
{   transmit                (0),
    receive                 (1),
    transmitAndReceive      (2)                    OPTIONAL?
    NumberOfChannels        INTEGER                DEFAULT 1 }
```

```
CSTACommonArgument3 ::= [APPLICATION 30] IMPLICIT SEQUENCE
{security                    [0]  IMPLICIT CSTASecurityData3  OPTIONAL
privateData                  [1]  IMPLICIT  SEQUENCE  OF  CSTAPrivateData3
   OPTIONAL }
CSTAPrivateData3 ::= CHOICE
{  string                    [0]  IMPLICIT OCTETSTRING
   privat                    [1]  IMPLICIT PrivatAPDU3 }
--The actual encoding is added here ,
--replacing NULL with another valid ASN.1 type.

PrivatAPDU3 ::= --snacc isPdu:"TRUE" – SEQUENCE
{  manufacture OBJECT IDENTIFIER
   ANY DEFINED BY manufacturer}
```

2.1 Operations

These operations are encapsulated in an Escape *service* and have the object identifier range 1.3.0012.1206.0 to 1.3.0012.1206.499.

2.1.1 Associate Data

2.1.1.1 Description

This service associates CSTA application Correlator Data, Account Code and/or Authorization Code information with a specified call. (not used – defined in ECMA phase 2 –)

2.1.1.2 ASN.1 encoding

```
    AssociateData           OBJECT-TYPE
        SYNTAX              AssociateData
        ACCESS              read-write
        STATUS              mandatory
        ::= { 1 3 0012 1206 0 }

AssociateData               ::= CHOICE
{   associateDataArgument   AssociateDataArgument,
    associateDataResult     AssociateDataResult }
AssociateDataArgument       ::= SEQUENCE
{   existingCall            ConnectionID,
    accountCode             [0] IMPLICIT AccountInfo      OPTIONAL,
    authCod                 [1] IMPLICIT AuthCode         OPTIONAL,
    correlatorData          [2] IMPLICIT CorrelatorData   OPTIONAL,
    extensions              CSTACommonArguments           OPTIONAL }

AssociateDataResult         ::= CHOICE
{   extensions              CSTACommonArguments,
    noData                  NULL }
```

2.1.1.3 Fields description

Refer to documents [1] and [2].

2.1.2 Set Date And Time

2.1.2.1 Description

This service is used to set the date and the local time on the switch. (not used from OmniPCX Enterprise release 4.1)

2.1.2.2 ASN.1 encoding

```
dateAndTimeSetting              OBJECT-TYPE
      SYNTAX                    DateAndTimeSetting
      ACCESS                    read-write
      STATUS                    mandatory
      ::= { 1 3 0012 1206 1 }

DateAndTimeSetting              ::= CHOICE
{   dateAndTimeSettingArgument  DateAndTimeSettingArgument,
    dateAndTimeSettingResult    DateAndTimeTimeSettingResult }

DateAndTi meSetti ngArgument    ::= SEQUENCE
{   dateAndTimeToSet            GeneralizedTime,
    extensions                  CSTACommonArguments   OPTIONAL }

DateAndTimeSettingResult        ::= CHOICE
{   extensions                  CSTACommonArguments,
    noData                      NULL }
```

2.1.2.3 Fields description

The *dateAndTime ToSet* field must have the following format (ASCII string as defined in ISO 2014 "Digital Representation of Dates"): YYYYMMDDhhmmss.s where:

YYYY is the year (4 characters), MM is the month (2 characters), DD is the day (2 chars),

hh is the hour (2 chars), mm are the minutes (2 chars), ss.s are the seconds.

For example:19961109100530.0 means local time 10 hours 5 minutes 30.0 seconds, the 9th of November 1996.

2.1.3 Host Information

2.1.3.1 Description

This service provided by the switch gives information about the system host to the CTI platform (reporting of a status like connected, disconnected...).

2.1.3.2 ASN.1 encoding

```
HostInformation                 OBJECT-          TYPE
      SYNTAX                    hostInformation
      ACCESS                    read-write
      STATUS                    mandatory
      ::= { 1 3 0012 1206 2 }

HostInformation                 ::=CHOICE
{   hostInformationArgument     HostInformationArgument,
    hostInformationResult       HostInformationResult }
HostInformationArgument         ::= SEQUENCE
{   hostIdentification          OCTET STRING,
    hostType                    HostType,
    hostStatus                  HostStatus,
    extensions                  CSTACommonArguments   OPTIONAL }

HostInformationResult           ::=CHOICE
{   extensions                  CSTACommonArguments,
    noData                      NULL }
The following types are used :
HostType                        ::= ENUMERATED
{   viola                       (0),
    generic-IVR                 (1),
    other                       (255) }
```

```
HostStatus                  ::= ENUMERATED
{   initializing            (0),-- formerly named connected
    enabled                 (1),-- formerly name disconnected
    normalSS                (2),-- formerly name disabled
    messagesLost            (3),-- formerly named enabled
    disabled                (4),-- formerly named initializing
    overloadImminent        (5),-- formerly named messages- lost
    overloadReached         (6),-- formerly named normal
    overloadRelieved        (7) -- formerly named overload- imminent}
```

2.1.3.3 Fields description

- *hostIdentification:* any string identifying the system host
- *hostType:* provides the host type
- *hostStatus:* host status reporting (connected, disconnected, enabled or disabled)

Note:

This operation was previously named *IVRInformation; IVRType* and *IVRStatus* types are now replaced *by HostType* and *HostStatus :*

2.1.4 Interrupt Transaction

2.1.4.1 Description

This service is used for a call in queue connected to an IVR port. The IVR does transactions and has to inform the distribution when these transactions are interruptible or not. With this information, the distribution is able to retrieve a call w/o cutting the current non interruptible transaction.

2.1.4.2 ASN.1 encoding

```
InterruptTransaction                OBJECT-TYPE
        SYNTAX                      InterruptTransaction
        ACCESS                      read-write
        STATUS                      mandatory
        ::= { 1 3 0012 1206 3 }
InterruptTransaction                ::= CHOICE
{   interruptTransactionArgument    InterruptTransactionArgument,
     interruptTransactionResult     InterruptTransactionResult }
InterruptTransactionArgument        ::= SEQUENCE
{   existingCall                    ConnectionID,
    interruptible                   BOOLEAN,
    extensions                      CSTACommonArguments   OPTIONAL }

InterruptTransactionResult ::=      CHOICE
{   extensions                      CSTACommonArguments,
    noData                          NULL }
```

2.1.4.3 Fields description

This operation was previously defined for "Internal use".

- *existingCall:* connection identifier of the call to retrieve
- *interruptible:* TRUE is the transaction is interruptible, FALSE in the opposite case

2.1.5 Set Device In Service

2.1.5.1 Description

This service is used to set a CSTA monitorable phone set in service or out of service.

2.1.5.2 ASN.1 encoding

```
setDeviceInService              OBJECT-TYPE
    SYNTAX                      SetDeviceInService
    ACCESS                      read-write
    STATUS                      mandatory
    ::= { 1 3 0012 1206 4 }

SetDeviceInService              ::= CHOICE
{   setDevice1nServiceArgument  SetDeviceInServiceArgument,
    setDeviceInServiceResult    SetDeviceInServiceResult }

SetDeviceInServiceArgument      ::=SEQUENCE
{   device                      DeviceID,
    setInService                BOOLEAN,
    extensions                  CSTACommonArguments  OPTIONAL}

SetDeviceInServiceResult        ::=CHOICE
{   extensions                  CSTACommonArguments,
    noData                      NULL }
```

2.1.5.3 Fields description

- *device:* device number
- *setinService:* TRUE means that the device will be set in service; FALSE will indicate out of service

2.1.6 Fast Data (not used – defined in ECMA phase 2 –)

2.1.6.1 Description

Bidirectional service used to set or to get data to or from a device (messages may be initiated by the switching function or by the computing function).

2.1.6.2 ASN.1 encoding

```
   fastData                OBJECT-TYPE
      SYNTAX               FastData
      ACCESS               read-write
      STATUS               mandatory
      ::= { 1 3 0012 1206 5 }

FastData                   ::=CHOICE
{  fastDataArgument        FastDataArgument,
   fastDataResult          FastDataResult }

FastDataArgument           ::= SEQUENCE
{  device                  CSTAObject,
   data                    IOData,
   dataPathDirection       DataPathDirection,      OPTIONAL
   dataPathType            DataPathType,           OPTIONAL
   extensions              CSTACommonArguments     OPTIONAL }

FastDataResult             ::=CHOICE
{  extensions              CSTACommonArguments,
   noData                  NULL }
The following types are used :
IOData                     ::= OCTET STRING
DataPathDirection::=       ENUMERATED
{  fromRequestor           (0),
   toRequestor             (1),
   biDirectional           (2) }
```

```
DataPathType                ::= ENUMERATED
{   text                    (0),
    digitalVoice            (1) }
```

2.1.6.3 Fields description

Refer to documents [1] and [2].

2.1.7 Supervisor Assist Request

2.1.7.1 Description

CCD feature allowing an agent to request support from a supervisor while engaged either in an ACD call or an external private call (if configured in the agent processing group).

2.1.7.2 ASN.1 encoding

```
SupervisorAssistRequest          OBJECT-TYPE
      SYNTAX                     SupervisorAssistRequest
      ACCESS                     read-write
      STATUS                     mandatory
      ::= { 1 3 0012 1206 6 }
SupervisorAssistRequest ::=      CHOICE
{  superviso rAssistRequest      Argument SupervisorAssistRequestArgument,
   supervisorAssistRequestResult SupervisorAssistRequestResult }

SupervisorAssistReq uestArgument ::=SEQUENCE
{  agentDevice                   DeviceID,
   extensions                    CSTACommonArguments  OPTIONAL }

SupervisorAssistRequestResult ::= CHOICE
{  supervisorDevice              DeviceID,
   extensions                    CSTACommonArguments,
   noData                        NULL }
```

2.1.7.3 Fields description

- *agentDevice:* agent number requesting the support of a supervisor
- *supervisorDevice:* first free supervisor number

Note:

A private event *SupervisorAssist Request* (see section 2.3.2) is generated in order to notify the assist request.

2.1.8 Supervisor Assist Cancel

2.1.8.1 Description

This service cancels a previously Supervisor Assist Request operation.

2.1.8.2 ASN.1 encoding

```
SupervisorAssistCancel          OBJECT-TYPE
      SYNTAX                    SupervisorAssistCancel
      ACCESS                    read-write
      STATUS                    mandatory
      ::= { 1 3 0012 1206 7 }
SupervisorAssistCancel ::=      CHOICE
{   supervisorAssistCancelArgument SupervisorAssistCancelArgument,
    supervisorAssistCancelResult   SupervisorAssistCancelResult }

SupervisorAssistCancelArgument  ::= SEQUENCE
{   requestingDevice            DeviceID,
    otherDevice                 DeviceID,
    extensions                  CSTACommonArguments  OPTIONAL }

SupervisorAssistCancelResult    ::=CHOICE
{   extensions                  CSTACommonArguments,
    noData                      NULL }
```

2.1.8.3 Fields description

- *requestingDevice:* indicates the device number that requests the cancel operation; it can be the agent or the supervisor
- *otherDevice:* indicates the Cher party (the supervisor or the agent)

Note:

A private event *SupervisorAssist Request* (see section 2.3.2) is generated in order to notify the cancel request.

2.1.9 Escape Register Request

2.1.9.1 Description

This service is used by the computing function to register to the switching function for receiving system services. The computing function is required to register for system services before it can receive any system service requests from the switching function.

2.1.9.2 ASN.1 encoding

```
ServiceRegisterRequest          OBJECT-TYPE
    SYNTAX                      ServiceRegisterRequest
    ACCESS                      read-write
    STATUS                      mandatory
    ::= { 1 3 0012 1206 8 }

ServiceRegisterRequest          ::= CHOICE
{   serviceReg isterReq uestArgument  ServiceRegisterRequestArgument,
    serviceRegisterRequestResult      ServiceRegisterRequestResult }

ServiceRegisterReq uestArgument ::= SEQUENCE
{   serviceVersionId            INTEGER,
    extensions                  CSTACommonArguments  OPTIONAL }

ServiceRegisterRequestResult    ::= CHOICE
{   serviceRegisterId           OCTET STRING,
extensions                      CSTACommonArguments,
noData                          NULL }
```

2.1.9.3 Fields description

- *serviceVersionid:* specifies the system service for which the computing function is requesting registration;
- value "10" means *Agent Logging Information* (see section 2.1.11);
- value “11” is related to listening services (DR-Link for NICE systems, see section 2.1.12);
- value “234” is related to listening services (DR-Link for other systems, see section 2.1.12);
- *serviceRegisterid:* specifies the system registration identifier for this registration (this value must be unique for the duration of the registration)

2.1.10 Escape Register Cancel

2.1.10.1 Description

This service is used to cancel a previous system registration. This request terminates the system registration and the computing function receives no further system service requests for that system registration once it receives the positive acknowledgement to the Escape Register Cancel request.

2.1.10.2 ASN.1 encoding

```
ServiceRegisterCancel          OBJECT-TYPE
    SYNTAX                     ServiceRegisterCance
    ACCESS                     read-write
    STATUS                     mandatory
    ::= { 1 3 0012 1206 9 }

ServiceRegisterCancel          ::=CHOICE
{   serviceRegisterCancelArgument  ServiceReg isterCancelArgument,
    serviceRegisterCancelResult    ServiceRegisterCancelResult }

ServiceReg isterCancelArgument ::=SEQUENCE
{   serviceRegisterId          OCTET STRING,
    extensions                 CSTACommonArguments  OPTIONAL }

ServiceRegisterCancelResult    ::=CHOICE
{   extensions                 CSTACommonArguments,
    noData                     NULL }
```

2.1.10.3 Fields description

- serviceRegisterid: specifies the system registration identifier for which the system registration is to be cancelled

2.1.11 Agent Logging Information

2.1.11.1 Description

This service is issued by the switching function to report the logging status of CCD agents without the requirement of monitoring the agents. It provides a CTI application with a mean for dynamically starting and stopping the monitoring on CCD agents.

2.1.11.2 ASN.1 encoding

```
AgentLoggingInformation          OBJECT-TYPE
    SYNTAX                       AgentLoggingInformation
    ACCESS                       read-write
    STATUS                       mandatory
    ::= { 1 3 0012 1206 10 }

AgentLoggingInformation          ::=CHOICE
{   agentLoggingInformationArgument   AgentLoggingInformationArgument,
agentLoggingInformationResult         AgentLoggingInformationResult
}
AgentLoggingInformationArgument  ::= SEQUENCE
{   serviceRegisterId            OCTET STRING,
    agentAction                  AgentAction,
    agentNumber                  DeviceID,
    proAcdNumber                 DeviceID,
    agentName                    IASString,              OPTIONAL,
    extensions                   CSTACommonArguments     OPTIONAL }

AgentLoggingInformationResult    ::= CHOICE
{   extensions                   CSTACommonArguments,
    noData                       NULL }
The following type is used :
AgentActio                        ::= ENUMERATED
{   loggedOff                    (0),
    loggedOn                     (1) }
```

2.1.11.3 Fields description

- *serviceRegisterid:* specifies the system registration identifier (allows to correlate the service with the original *Register Request* operation)
- *agentAction:* logon or logoff operation
- *agentNumber:* CCD agent number
- *proAcdNumber:* pro-ACD number of the CCD agent
- *agentName:* CCD agent name

Note:

To access this service, the application has to register to the switch using the private service Escape *Register Request*.

2.1.12 Fields description

Parameters *presenceTimer, silenceTimer* and *tonality* allow to define the "beep" tone characterized by a square pulse signal.

- *bippedDevice:* device on which the "beep" tone is received
- *tonality:* frequency to connect; designates a time slot number defined in the OmniPCX Enterprise tonalities table which is specific for each country (in practice, only values 21 to 30 are usable)
- *presenceTimer:* presence timer of the pulse (value specified in step of 10ms)
- *silenceTimer:* silence timer of the pulse (value specified in step of 10ms); for sending only one "beep" tone, set this value to 0

2.1.13 Roaming DECT Status

2.1.13.1 Description

Roaming DECT localization enables CTI applications to track, in an OmniPCX Enterprise environment, a DECT handset that would enter roaming phase (see also section 2.3.3).

This service allows to request localization information and can be executed either if the device is in the idle state or not.

2.1.13.2 ASN.1 encoding

```
roamingDectStatus                OBJECT-TYPE
      SYNTAX                     RoamingDectStatus
      ACCESS                     read-write
      STATUS                     mandatory
      ::= { 1 3 0012 1206 14 }
RoamingDectStatus                ::= CHOICE
{  roamingDectStatusArg u ment   [0] IMPLICIT RoamingDectStatusArgument,
   roamingDectStatusResult       [1 ] IMPLICIT RoamingDectStatusResult }

RoamingDectStatusArgument        ::= SEQUENCE
{  deviceNumber                  DeviceID,
   extensions                    CSTACommonArguments   OPTIONAL }

RoamingDectStatusResult          ::= CHOICE
{  moved                         BOOLEAN,
   newNode                       [10] IMPLICIT INTEGER   OPTIONAL,
   subNetworkNb                  [11] IMPLICIT INTEGER   OPTIONAL,
   shellNumber                   DeviceID                OPTIONAL,
   extensions                    CSTACommonArguments     OPTIONAL }
```

2.1.13.3 Fields description

- *deviceNumber:* directory number of the DECT set to query
- *moved:* TRUE means that the device has moved from one OmniPCX Enterprise node to an other one (FALSE if not); in this case, fieldsnewNode, *subNetworkNb* and *shellNumberwill* befilled (NULL values otherwise)
- *newNode:* new destination node
- *subNetworkNb:* sub-network where the set is
- *shellNumber:* shell directory number of the device on the new node

Note:

The service is only available on the home node, that is the OmniPCX Enterprise node on which the DECT set has been registered.

2.1.14 Permanent Listening

2.1.14.1 Activation Description

Allows a CCD supervisor, which is in the idle state, to activate a permanent listening on a CCD agent phone set. Agent can be either in idle state or in conversation state.

2.1.14.2 ASN.1 encoding

```
permanentListeningActivation        OBJECT-TYPE
    SYNTAX                          PermanentListeningActivation
    ACCESS                          read-write
    STATUS                          mandatory
    ::= { 1 3 0012 1206 15 }

PermanentListeningActivation        ::= CHOICE
{permanentListeningActivationArgument   PermanentListeningActivationArgument,
permanentListeningActivationResult      PermanentListeningActivationResult }

PermanentListeningActivationArgument ::=            SEQUENCE
{   requestingDevice                DeviceID,
    listenAgentDevice               DeviceID,
    extensions                      CSTACommonArguments   OPTIONAL }

PermanentListeningActivationResult ::= CHOICE
{   extensions                      CSTACommonArguments   OPTIONAL,
    noData                          NULL }
```

2.1.14.3 Fields description

- *requestingDevice:* CCD supervisor number
- *listenAgentDevice:* CCD agent number

Note:

This request is canceled either by using the CSTA Clear Connection service or through the softkey configured on the supervisor phone set.

2.1.15 Call Center Treatment request

2.1.15.1 Description

This service allows a CCD Operator to Start or Stop a specific Call Center treatment for CCO (from release R4.1), CCE and CCW (from release R4.1.1) applications

2.1.15.2 ASN.1 encoding

```
ServiceEscapeCCTreatmentRequest OBJECT-TYPE
        SYNTAX                  ServiceEscapeCCTreatmentRequest
        ACCESS                  read-write
        STATUS                  mandatory
        ::= { 1 3 0012 1206 16 }

--Operation Service Call Center Treatment in Escape Service--
ServiceEscapeCCTreatmentRequest                 ::= CHOICE
{   ServiceEscapeCCTreatmentRequestArgument     [0] IMPLICIT
    ServiceEscapeCCTreatmentRequestArgument,
    ServiceEscapeCCTreatmentRequestArgument     [1] IMPLICIT
    ServiceEscapeCCTreatmentRequestArgument,}

ServiceEscapeCCTreatmentRequestArgument         ::= SEQUENCE
{   agentNumber                 DeviceID,
    actionType                  BOOLEAN,
    applicationType             CCTreatmentType
    extensions                  CSTACommonArguments  OPTIONAL
}
--Result for operation – Service Call Center Treatment Request--
ServiceEscapeCCTreatmentRequestResult           ::= SEQUENCE
{   globalCallID                GlobalCallID        OPTIONAL,
    extensions                  CSTACommonArguments  OPTIONAL }
```

2.1.15.3 Fields description

- requestingDevice:
- Note

2.1.16 Set Message Waiting Indicator

2.1.16.1 Description

Allows to control the status (on/off) of the message waiting indicator at a specified device. This feature is typically used to notify a user (through the lamp on the phone set) when messages are available.

2.1.16.2 ASN.1 encoding

```
setMessageWaitingIndicator      OBJECT-TYPE
    SYNTAX                      SetMessageWaitingIndicator
    ACCESS                      read-write
    STATUS                      mandatory
    ::= { 1 3 0012 1206 17 }

SetMessageWaitingIndicator      ::= CHOICE
{   setMessageWaitingIndicatorArgument   SetMessageWaitingIndicatorArgument,
setMessageWaitingIndicatorResult         SetMessageWaitingIndicatorResult }

SetMessageWaitingIndicatorArgument  ::= SEQUENCE
{   device                          DeviceID,
    messageWaitingOn                BOOLEAN,
    extensions                      CSTACommonArguments    OPTIONAL }

SetMessageWaitingIndicatorResult    ::= CHOICE
{   extensions                      CSTACommonArguments  OPTIONAL,
    noData                          NULL }
```

2.1.16.3 Fields description

- *device*: device's physical element
- *messageWaitingOn*: TRUE means message waiting on and FALSE means message waiting off

Note:

User device must be connected to a VPS voice mail system.

2.1.17 Start Data Collection

2.1.17.1 Description

This service is used to collect information such as DTMF pulse digits from a connection at a specific device.

Note:

This escape service is based on equivalent the ECMA 3 definition. For more information, refer to Standard ECMA–285 (Protocol for Computer–Supported Telecommunications Applications- Phase III) (2nd Edition – June 2000)

2.1.17.2 ASN.1 encoding

```
ServiceEscapeStartDataCollection  OBJECT-TYPE
        SYNTAX                    ServiceEscapeStartDataCollection
        ACCESS                     read-write
        STATUS                    mandatory
        ::= { 1 3 0012 1206 18 }

--Operation Start Data Collection in Escape Service--
ServiceEscapeStartDataCollection  ::= CHOICE
{   serviceEscapeStartDataCollectionArgument
        [0]IMPLICIT serviceEscapeStartDataCollectionArgument,
    serviceEscapeStartDataCollectionResult
        [1]IMPLICIT serviceEscapeStartDataCollectionResult}

--Argument for operation - Service Start Data Collection --

DataCollType3::=              ENUMERATED
{   digits                    (0),
    telTones                  (1) }

DigitsReportingCriteria3      ::= SEQUENCE
{   numberChars               [0] IMPLICIT INTEGER     OPTIONAL,
    flushChar                 (A5String (SIZE(1_1))    OPTIONAL,
    timeout                   [1] IMPLICIT INTEGER     OPTIONAL }
```

```
StartDataCollection Argument        ::= SEQUENCE
{   object                          CallObject3,
dataCollType                        DataCollType3               OPTIONAL,
digitalReportingCriteria            DigitalReportingCriteria3   OPTIONAL,
extensions                          CSTACommonArguments3        OPTIONAL }

--Result for operation – Service Start Data Collection—
DcollCrossRefID3                    ::= OCTET STRING (SIZE(0..4))

StartDataCollection Result          ::= SEQUENCE
{   DcollCrossRefID                 DcollCrossRefID3,
    extensions                      CSTACommonArguments3        OPTIONAL }
```

2.1.18 Stop Data Collection

2.1.18.1 Description

The Stop Data escape service terminates an existing data collection.

Note:

This escape service is based on equivalent the ECMA 3 definition. For more information, refer to Standard ECMA–285 (Protocol for Computer–Supported Telecommunications Applications- Phase III) (2nd Edition – June 2000)

2.1.18.2 ASN.1 encoding

```
ServiceEscapeStartDataCollected   OBJECT-TYPE
        SYNTAX                    ServiceEscapeStartDataCollected
        ACCESS                    read-write
        STATUS                    mandatory
        ::= { 1 3 0012 1206 19 }
--Argument for operation - Service Stop Data Collection --

StopDataCollection Argument       ::= SEQUENCE
{   datacollCrossRefID            datacollCrossRefID 3        ,
    extensions                    CSTACommonArguments3        OPTIONAL }

--Result for operation – Service Stop Data Collection--

StopDataCollection Result ::=     CHOICE
{   extensions                    CSTACommonArguments3 ,
noData                            NULL }
```

2.1.18.3 Fields description

- requestingDevice:
- Note

2.1.19 Data Collected

2.1.19.1 Description

The Data Collected escape service sends data that is received over a connection to the computing function.

Note:

This escape service is based on equivalent the ECMA 3 definition. For more information, refer to Standard ECMA–285 (Protocol for Computer–Supported Telecommunications Applications- Phase III) (2nd Edition – June 2000)

2.1.19.2 ASN.1 encoding

```
ServiceEscapeDataCollected      OBJECT-TYPE
      SYNTAX                    ServiceEscapeDataCollected
      ACCESS                    read-write
      STATUS                    mandatory
      ::= { 1 3 0012 1206 20 }

ServiceEscapeDataCollected      ::= CHOICE
{   serviceEscapeStartDataCollectedArgument     DataCollectedArgument,
    serviceEscapeStartDataCollectedResult       DataCollectedResult }
--Argument for operation - Service Start Data Collected --

DigitsData3                     ::=SEQUENCE
{   digits detected             [(A5String (SIZE(0..64)),
    digitsDuration              [0] IMPLICIT SEQUENCE OF INTEGER
    OPTIONAL,
    digitsPauseDuration         [1] IMPLICIT INTEGE
    OPTIONAL }

DigitsData3                     ::= ENUMERATED
{   flushCharReceived           (0),
    charCountReached            (1),
    timeout                     (2),
    sfTerminated                (3)}
```

```
DataCollectArgument,            ::= SEQUENCE
{   dcollCrossRefID             DcollCrossRefID3,
    digitsData                  [0] IMPLICIT DigitsData3,
    OPTIONAL
    telTonesData                [1] IMPLICIT TelTonesData3
    OPTIONAL
    connectioninformation       [2] IMPLICIT ConnectionInformation3
    OPTIONAL
    dcollCause                  DcollCause3
    OPTIONAL
    extensions                  CSTACommonArguments3
    OPTIONAL }

--Result for operation – Service Data Collected--

StartDataCollectedResult        ::= CHOICE
{   extensions                  CSTACommonArguments3,
    noData                      NULL }
```

2.1.19.3 Fields description

- requestingDevice:
- Note

2.1.20 Generate Telephony Tones

2.1.20.1 Description

The Generate Telephony Tones escape service causes a telephony tone such as a busy or a ringback to be sent on behalf of a collection in a call tone.

For RSI, this service is extended to play music and announcement features.

Note:

This escape service is based on equivalent the ECMA 3 definition. For more information, refer to Standard ECMA–285 (Protocol for Computer–Supported Telecommunications Applications- Phase III) (2nd Edition – June 2000)

2.1.20.2 ASN.1 encoding

```
ServiceEscapeGenerateTelTones   OBJECT-TYPE
        SYNTAX                  ServiceEscapeGenerateTelTones
        ACCESS                  read-write
        STATUS                  mandatory
        ::= { 1 3 0012 1206 21 }

ServiceEscapeGenerateTelTones   ::= CHOICE
{serviceEscapeGenerateTelTonesArgument
    GenerateTelephonyTonesArgument3,
serviceEscapeGenerateTelTonesResult     GenerateTelephonyTonesResult3,}

--Argument for operation - Service GenerateTelephony Tones --
TelephonyTones3 ::=             ENUMERATED
{   beep                        (0),
    billing                     (1),
    timeout                     (2),
    carrier                     (3),
    confirmation                (4),
    dial                        (5),
    faxCNG                      (6),
    hold                        (7),
    howler                      (8),
    intrusion                   (9),
    modemCNG                    (10),
    park                        (11),
    recordWarning               (12),
```

```
    reorder                         (13),
    ringback                        (14),
    silence                         (15),
    sitVC                           (16),
    sitIC                           (17),
    sitRO                           (18),
    sitNC                           (19),
    switchSpec0                     (20),
    switchSpec1                     (21)
    SwitchSpec100                   (120) }
GenerateTelephonyTonesArgument ::= SEQUENCE
{   connectionToSendTone            ConnectionID3,
    toneToSend                      TelephonyTone3
    toneDuration                    INTEGER                     OPTIONAL,
    extensions                      CSTACommonArguments3        OTIONAL }

--Result for operation – GenerateTelephonyTonesArgument --

GenerateTelephonyTonesResult3   ::= CHOICE
{   extensions                      CSTACommonArguments3,
    noData                          NULL }
```

2.1.20.3 Fields description

- requestingDevice:
- Note

2.1.21 Cancel Telephony Tones

2.1.21.1 Description

The Cancel Telephony Tones escape service cancels a telephony tone that is being sent on a connection by the Generate Telephony Tones Service.

Note:

This escape service is based on equivalent the ECMA 3 definition. For more information, refer to Standard ECMA–285 (Protocol for Computer–Supported Telecommunications Applications- Phase III) (2nd Edition – June 2000)

2.1.21.2 ASN.1 encoding

```
ServiceEscapeCancelTelTones      OBJECT-TYPE
        SYNTAX ServiceEscapeCancelTelTones
        ACCESS read-write
        STATUS mandatory
        ::= { 1 3 0012 1206 22 }
ServiceEscapeCancelTelTone       ::= CHOICE
{   serviceEscapeCancelTelTonesArgument   CancelTelephonyTonesArgument3,
    serviceEscapeCancelTelTonesResult     CancelTelephonyTonesResult3 }

--Argument for operation - Service Cancel Telephony Tones --
CancelTelephonyTonesArgument3  ::= SEQUENCE
{   connectionToStopTone       ConnectionID3,
    extensions                 CSTACommonArguments3 OPTIONAL }

--Result for operation – Service Cancel Telephony Tones --
CancelTelephonyTonesResult3    ::= CHOICE
{   extensions                 CSTACommonArguments3,
    noData                     NULL }
```

2.1.21.3 Fields description

- requestingDevice:
- Note

2.1.22 Other Operations

2.1.22.1 Description

Allows definition of non Alcatel-Lucent private operations.

2.1.22.2 ASN.1 encoding

```
OtherOperation                  OBJECT-TYPE
        SYNTAX OtherOperation
        ACCESS read-write
        STATUS mandatory
        ::= { 1 3 0012 1206 499 }

OtherOperation                  ::= SEQUENCE
{   operation Identifier        [0] IMPLICIT OCTET STRING,
    CHOICE
{    operationArgument          [0] OCTET STRING,
    operationResult             [1 ] OCTET STRING
    extensions                  CSTACommonArguments     OPTIONAL }
```

2.2 Private data

These data have the object identifier range 1.3.0012.1206.500 to 1.3.0012.1206.999 . (not used – defined in ECMA phase 2)

2.2.1 Description

Contains information supplied by the computing information application.

2.2.1.1 ASN.1 encoding

```
correlatorData                  OBJECT-TYPE
    SYNTAX CorrelatorData
    ACCESS read-write
    STATUS mandatory
    ::= { 1 3 0012 1206 500}
```

2.2.1.2 Fields description

Refer to documents [1] and [2].

2.2.2 Private Errors

2.2.2.1 Description

Provides diagnostic error definitions. (not used – defined in ECMA phase 2)

2.2.2.2 ASN.1 encoding

```
privateErrors                   OBJECT-TYPE
      SYNTAX PrivateErrors
      ACCESS read-write
      STATUS mandatory
      ::= { 1 3 00121206 501}
PrivateErrors                   ::= CHOICE
{   operationalErrors           [1]IMPLICIT PrivateOperations,
    unspecifiedErrors           [7]IMPLICIT NULL,
    otherErrors                 [8]IMPLICIT OCTET STRING }

PrivateOperations::=            ENUMERATED
{   invalidCSTAApplicationCorrelator (20),
    invalidAccountCode          (21),
    invalidAuthorizationCode    (22) }
```

2.2.2.3 Fields description

Refer to documents [1] and [2].

2.2.3 CCD Pilot Expected Waiting Time

2.2.3.1 Description

CCD feature allowing to report, in the Delivered event and in the acknowledgment of a Query operation to a Pilot number, the expected waiting time and the queue saturation flag.

2.2.3.2 ASN.1 encoding

```
aCDWaitingTime              OBJECT-TYPE
    SYNTAX                  ACDWaitingTime
    ACCESS                  read-write
    STATUS                  mandatory
    ::={ 1 3 0012 1206 502 }

ACDWaitingTime              ::= SEQUENCE
{   waitingTime             INTEGER,
    saturation              BOOLEAN }
```

2.2.3.3 Fields description

- *waitingTime:* indicates the estimated waiting time in seconds
- *saturation:* TRUE means queue saturated

2.2.4 Service Option

2.2.4.1 Description

Used to activate a specific option in services like Make Call, Consultation Call, Single Step Conference, Query on Pilot and Set Feature Logged On.

2.2.4.2 ASN.1 encoding

```
CstaServiceOption                       OBJECT-TYPE
    SYNTAX                              CstaServiceOption
    ACCESS                              read-write
    STATUS                              mandatory
    ::= { 1 3 0012 1206 503 }
  --Alcatel-Lucent Service Option description--
CstaServiceOption ::=                   BIT STRING
{   callProgressToneInhibition
    CallProgressToneInhibition          (0)
    holdToneInhibition prioritaryTransfer (1)
    priorityTransfer                    (2)
    autoOriginate                       (3)
    supervisorCall                      (4)
    supervisorStepIn                    (5)
    supervisedTransfer                  (6)
    headsetMode                         (7)
    backupRouting                       (8) }
```

2.2.4.3 Fields description

- *callProgressToneInhibition:* suppresses the progress tone (used in Make Call and Consultation Call)
- *holdToneInhibition:* suppresses the hold tone between an IVR access and an external customer during the Consultation Call service performed by the IVR
- *prioritaryTransfer:* makes a priority transfer (used in Consultation Call and Query to a Pilot number)
- *autoOriginate:* allows an application to auto-originate a call from a device (used in Make Call); this option is also available for a virtual device
- *supervisorCall:* allows a CCD agent to call a supervisor for assistance, in either idle or conversation state (used in Make Call and Consultation Call)
- *supervisorStepIn:* allows a CCD supervisor to talk with an agent without being heard from the calling party (used in Single Step Conference)

- *supervisedTransfer:* allows to consult a CCD pilot with immediate distribution to an agent if any available (used in Consultation Call); offers a mean to transfer a call from a CCD agent to another agent with respect to agent's availability (if no agent is available, the CCD pilot consultation behaves as in OmniPCX Enterprise release 2.1)
- *headsetMode*: allows a CCD agent to select the headset mode at logon (used in Set Feature Logged On)
- *backupRouting* enables RSI routing Point as "backup" used in Set Feature routing enable)

Note:

In the Consultation Call service, it is not possible to simultaneously use two or more of the following options: *prioritaryTransfer, supervisedTransfer* and *supervisorCall.*

2.2.5 Network Time Slot

2.2.5.1 Description

Provides the network time slot (ISDN B channel) used by the external call that was transferred or conferenced. This data allows to manage trunk side recording on ISDN when a monitored device is conferenced in or receives a transferred external call.

The data is reported in the Delivered, Queued, Transfered and Conferenced events and is only available when combining a single external call with a local call.

2.2.5.2 ASN.1 encoding

NetworkTimeSlot	OBJECT-TYPE
SYNTAX I	NTEGER
ACCESS	read-write
STATUS	mandatory
::={ 1 3 0012 1206 504 }	

2.2.6 Not Ready Context

2.2.6.1 Description

Used, in the CCD Agent Assignment Information private data (see section 2.2.7), to specify a particular flavor for the Not Ready state.

2.2.6.2 ASN.1 encoding

```
notReadyContext                 OBJECT-TYPE
    SYNTAX                      NotReadyContext
    ACCESS read-write
    STATUS                      mandatory
    ::={ 1 3 0012 1206 505 }
The following type is used :
NotReadyContext ::=             ENUMERATED
{   forcedPause                 (0),
    other                       (255) }
```

2.2.7 CCD Agent Assignment Information

2.2.7.1 Description

CCD feature allowing, in the Query Agent State function result, to report information depending on the agent state (logged off, pre-assigned, ready, not ready, wrap up, pause, busy).

2.2.7.2 ASNA encoding

```
QueryAgentInformation          OBJECT-TYPE
      SYNTAX                   QueryAgentInformation
      ACCESS                   read-write
      STATUS                   mandatory
      ::={ 1 3 0012 1206 506 }

QueryAgentInformation          ::= SEQUENCE
{  agentPreAssigned            BOOLEAN                 OPTIONAL,
   agentGroup                  AgentGroup              OPTIONAL,
   notReadyContext             NotReadyContext  OPTIONAL }
The following type is used :
NotReadyContext ::=            ENUMERATED
{  forcedPause                 (0),
   other                       (255) }
```

2.2.7.3 Fields description

- *agentPreAssigned:* TRUE means agent is pre-assigned
- *agentGroup:* CCD group in which the agent is assigned
- *notReadyContext* indicates a particular flavor for the Not Ready state

2.2.8 CCD Pilot Status

2.2.8.1 Description

Used in the Delivered event and in the Query Pilot function result to a Pilot number when consulting a CCD Pilot.

2.2.8.2 ASN.1 encoding

```
pilotTransferInfo                   OBJECT-TYPE
      SYNTAX                        PilotTransferInfo
      ACCESS                        read-write
      STATUS                        mandatory
      ::={ 1 3 0012 1206 507 }
PilotTransferInfo
{  possibleTransfer                 BOOLEAN,
   pilotStatus                      PilotStatus }
The following type is used :
PilotStatus                         ::= ENUMERATED
{  open                             (0),
   blocked                          (1),
   blocked-on-rule                  (2),
   blocked-on-blocked-rule          (3),
   general-forwarding               (4),
   general-forwarding-on-rule       (5),
   blocked-on-general-forwarding-rule (6),
   other                            (255) }
```

2.2.8.3 Fields description

- *possibleTransfer:* TRUE means transfer to a Pilot number possible.
- *pilotStatus:* state of the CCD Pilot

2.2.9 CCD Delivered Call Information

2.2.9.1 Description

Used in the Delivered and Established events to report CCD distribution information (especially the real waiting time of the call).

2.2.9.2 ASN.1 encoding

```
aCDDistributionInfo             OBJECT-TYPE
      SYNTAX                    ACDDistributionInfo
      ACCESS                    read-write
      STATUS                    mandatory
      :={ 1 3 0012 1206 508 }

ACDDistributionInfo ::=         SEQUENCE
{   waitingTime                 INTEGER,
    globalWaitingTime           INTEGER,
    agentGroup                  AgentGroup          OPTIONAL }
```

2.2.9.3 Fields description

- *waitingTime:* waiting time in the queue
- *globalWaitingTime:* global waiting time in the distribution
- *agentGroup:* processing group that distributes the call

2.2.10 Interactive Queuing Delivered Call Information

2.2.10.1 Description

CCD feature. Appears in the Delivered event when a call in queue is presented to an IVR port.

2.2.10.2 ASN.1 encoding

```
IVRInteractiveQueueing          OBJECT-TYPE
    SYNTAX                      IVRInteractiveQueueing
    ACCESS                      read-write
    STATUS                      mandatory
    ::= { 1 3 0012 1206 509 }

IVRInteractiveQueueing          ::= SEQUENCE
{   pilotNumber                 SubjectDeviceID,
    guideCurrentLevel           INTEGER }
```

2.2.10.3 Fields description

- *pilotNumber:* called Pilot number
- *guideCurrentLevel:* guide level when entering in interactive queuingphase

Note:

This data was previously defined for "internal use".

2.2.11 Interactive Queuing Reconnect Guide Level

2.2.11.1 Description

Used in the Clear Connection service (CCD feature). If a call in queue is connected to an IVR port and if the IVR ends its transaction, the connection is cleared and can provide distribution with the next waiting level to connect the call.

2.2.11.2 ASN.1 encoding

iVRNextLevel	OBJECT-TYPE
SYNTAX	INTEGER
ACCESS	read-write
STATUS	mandatory
::= { 1 3 0012 1206 510 }	

Note:

This data was previously defined for "internal use".

2.2.12 Withdrawal Type

2.2.12.1 Description

CCD feature allowing a CTI application to qualify the withdrawal state (values 1 to 9) of a CCD agent (e.g. coffee break) according to the agent's processing group management. This data is used in the Not Ready event and in the Set Feature Not Ready service.

2.2.12.2 ASN.1 encoding

```
notReadyActivation          OBJECT-TYPE
    SYNTAX                  NotReadyActivation
    ACCESS                  read-write
    STATUS                  mandatory
    ::={ 1 3 0012 1206 511 }

NotReadyActivation          ::= INTEGER
```

Note:

Using the value 0 in the Set Feature Not Ready service will force the pause state when the agent is in the wrap-up state.

2.2.13 Party Name

2.2.13.1 Description

Used to convey calling or called/alerting party name in events Delivered, Queued, Logged On and Logged Off.

2.2.13.2 ASN.1 encoding

```
partyName                   OBJECT-TYPE
    SYNTAX                  PartyName
    ACCESS                  read-write
    STATUS                  mandatory
    ::= { 1 3 0012 1206 512 }

PartyName                   ::= SEQUENCE
{   callingPartyName        [0] IMPLICIT IA5String      OPTIONAL,
    alertingPartyName       [1] IMPLICIT IA5String      OPTIONAL }
```

2.2.13.3 Fields description

- *callingPartyName:* provides the name associated with the calling party; it is the display narre that might be different from the phonebook or data base name
- *alertingPartyName:* called or alerting party name

2.2.14 Requesting Device

2.2.14.1 Description

Used in the Logged On/Logged Off events and in the Query Agent State result to report the directory number of the pro-ACD device.

In the Clear Connection service, allows to drop the added party from a third-partyconference, leaving the original parties in conversation. The requesting device is used to designate the conferencing device (master).

The *Requesting Device* data is also used for Alcatel-Lucent 4980 Nomadic support. A functionality of this application is to have remotely, for a Tele-worker using a GSM set, a transparent access to voice and data services from his company.

A Make Call service is initiated from a virtual set to the GSM number and the private data is used to indicate the set that will be charged for this call (e.g. the Tele-worker's UA set in the company).

2.2.14.2 ASN.1 encoding

```
requestingDevice            OBJECT-TYPE
    SYNTAX                  RequestingDevice
    ACCESS                  read-write
    STATUS                  mandatory
    ::={ 1 3 0012 1206 513 }
RequestingDevice            ::= DeviceID
```

Note:

For the Alcatel-Lucent 4980 Nomadic support, the CSTA event flow report has no specificity, that is the calling party is the virtual set and the called party is the GSM number (only external numbers via TO/T2 or ABC-F accesses are supported).

2.2.15 Rerouted Call Indication

2.2.15.1 Description

CCD feature indicating, in the Delivered event reported on a CCD Pilot, if an ACD call is rerouted or not in the network.

2.2.15.2 ASN.1 encoding

AcdNetworkRerouted	OBJECT-TYPE
SYNTAX	BOOLEAN
ACCESS	read-write
STATUS	mandatory
::= { 1 3 0012 1206 514 }	

Note:

acdNetworkRerouted set to TRUE means that the call has been rerouted.

2.2.16 Host Identification

2.2.16.1 Description

Used to specify a host identification in a System Status operation (at the moment, for IVR synchronization).

2.2.16.2 ASN.1 encoding

hostIvrIdentification	OBJECT-TYPE
SYNTAX	OCTET STRING
ACCESS	read-write
STATUS	mandatory
::= { 1 3 0012 1206 515 }	

2.2.17 Supervised Transfer

2.2.17.1 Description

Allows a CTI application to select the transfer mode of a given CCD Pilot on a per call basis. In the Consultation Call service, this private data forces to YES or NO the CCD Pilot consultation managed by the switch.

This data is also used:

- in the Delivered event (on the agent) to get the type of transfer effectively realised (accompanied or not);
- in the Query Device Information service result to a Pilot number to provide the value (YES or NO) managed at Pilot level.

2.2.17.2 ASN.1 encoding

```
SupervisedTransfer      OBJECT-TYPE
    SYNTAX              BOOLEAN
    ACCESS              read-write
    STATUS              mandatory
    ::= { 1 3 0012 1206 516 }
```

Note:

This data can be in conflict with the Service Option *supervisedTransfer* (see section 2.2.4). The following table enumerates the possible interaction cases:

DATA SERVICE OPTION ACCOMPANIED TRANSFER VS BOOLEAN DATA ACCOMPANIED TRANSFER	RESULT
Service Option not specified / accompanied transfer set to YES	Accompanied transfer will be realized
Service Option not specified / accompanied transfer set to NO	No accompanied transfer (by-passes the Pilot configuration)
Service Option not specified / accompanied transfer not specified	Usage of the Pilot configuration
Service Option specified / accompanied transfer set to YES	Accompanied transfer will be realized
Service Option specified / accompanied transfer set to NO	No accompanied transfer (by-passes the Pilot configuration)
Service Option specified / accompanied transfer not specified	Accompanied transfer will be realized

2.2.18 ACR attributes list

Used in the Delivered event on an IVR port (in interactive queuing or not) to give the list of sjkills associated to the call.

In the consultation call services on CCD Pilot (performed by IVR posts or Call Center virtual sets) or in the associate data service, allows the transmission of skills for CCD distribution purposes.

In the query service on CCD Pilot, it allows the retrieval of CCD distribution information according to the call profile.

In the Make Call service on CCD agent, it allows outgoing call type definition (CCO, CCE, CCW,…) according to the given list of skills.

```
acrAttributeList                OBJECT-TYPE
      SYNTAX                    ACRAttributeList
      ACCESS                    read-write
      STATUS                    mandatory
      ::={ 1 3 0012 1206 518 }

   --Alcatel-Lucent attributes list description--
ACRAttributeList                ::= SEQUENCE
{  acrAgentNumber               DeviceID          OPTIONAL
   attributeList                [2]IMPLICIT SEQUENCE OF Element
   OPTIONAL }
   Argument for Alcatel-Lucent attributes list description--
Element                         ::=SEQUENCE
{  skillNumber                  INTEGER
   acrStatus                    BOOLEAN
expertEvaluationLevel           INTEGER }
```

2.2.19 CCD Treatment Type

2.2.19.1 Description

In OmniPCX Enterprise release 4.1, this data is used to report, in the Working After Call event, and in the Call Center Treatment Request the treatment type performed by the agent (e-mail call, web call or call related to an outbound campaign).

In OmniPCX Enterprise release 4.1.1, this data is also used in events Delivered (see description above) and Ready (for specific CCD behaviors).

2.2.19.2 ASN.1 encoding

```
cCTreatmentType          OBJECT-TYPE
    SYNTAX               CCTreatmentType
    ACCESS               read-write
    STATUS               mandatory
    ::={ 1 3 0012 1206 519 }
```

Definition of *CCTreatmentType* in OmniPCX Enterprise release 4.1 is as follows:

```
CCTreatmentType ::=      ENUMERATED
{   cCEApplication       (0),
    cCOApplication       (1),
    cCWApplication       (2),
    other                                (255) }
```

Definition of *CCTreatmentType* in OmniPCX Enterprise release 4.1.1 is as follows:

```
CCTreatmentType ::=      ENUMERATED
{   cCEApplication       (0),
    cCOApplication       (1),
    cCWApplication       (2),
    cBKApplication       (3),
    cCETransfer          (4),
    cCEndPauseWithdraw   (5),
other                    (255) }
```

2.2.20 Secrecy Identity List

2.2.20.1 Description

Used in all call control events to report parties identity involved in a call if secrecy identity feature has been activated.

This data provides a list of device identifiers and is event related. Display order in the list corresponds to the display order of the fields in CSTA events.

Example: in the Originated event, the list respectively provides the calling device, the called device and the originating device.

In a snapshot device result on behalf of a "trunk- trunk conference"(digicom7) type virtual set, the SecretIdentity List gives the directory numbers of the remote device linked with each Connection Id

2.2.20.2 ASN.1 encoding

secretIdentiteList	OBJECT-TYPE
SYNTAX	SecretIdentiteList
ACCESS	read-write
STATUS	mandatory
::={ 1 3 0012 1206 520 }	

SecretIdentiteList ::= SEQUENCE OF DeviceID

Note:

Secrecy identity is activated through CSTA management on the switch side.

2.2.21 Global Call Identifier

2.2.21.1 Description

Allows a network wide call tracking by provision of a unique call linkage identifier. This identifier is provided in addition to the current CSTA call identifier and is unique across the OmniPCX Enterprise network.

The global call linkage identifier is provided in:

the acknowledgement of CSTA services creating or modifying call identifiers (Make Call, Consultation Call, Transfer Call and Conference Call);

all CSTA call control events.

2.2.21.2 ASN.1 encoding

globalCallId	OBJECT-TYPE
SYNTAX	GlobalCallId
ACCESS	read-write
STATUS	mandatory
:: = { 1 3 0012 1206 521 }	
GlobalCallId	::= OCTET STRING

Note:

This feature requires ABC–F2 protocol and homogeneous OmniPCX Enterprise switch versions and is available from OmniPCX Enterprise release 4.1.1.

2.2.22 Old Global Call Identifier Reminder

2.2.22.1 Description

Refer to section 2.2.20 for the goal of the global call linkage identifier.

This data is used in CSTA Transferred and Conferenced events to indicate that call linkage information has changed following a transfer or a conference operation.

2.2.22.2 ASN.1 encoding

```
oldGlobalCallId          OBJECT-TYPE
    SYNTAX               GlobalCallId
    ACCESS               read-write
    STATUS               mandatory
    ::={ 1 3 0012 1206 522 }
GlobalCallId             ::= OCTET STRING
```

Note:

This feature requires ABC–F2 protocol and homogeneous switch versions and is available from OmniPCX Enterprise release 4.1.1.

2.2.23 National/International Indicator

2.2.23.1 Description

Provides a mean to discriminate international from national ISDN calls in Delivered, Established and Queued events.

If the ISDN information is present, this data uses the *explicitPublic* field of the number (see CSTA definitions below) to provide international/national indication.

2.2.23.2 ASN.1 encoding

```
nationalIndication              OBJECT-TYPE
    SYNTAX                      NationalIndication
    ACCESS                      read-write
    STATUS                      mandatory
    ::={ 1 3 0012 1206 523 }
NationalIndication ::= ExtendedDeviceID
```

ExtendedDeviceID is defined as follows in the CSTA standard:

```
ExtendedDeviceID                ::= CHOICE
{  device identifier            DeviceID,
   implicitPublic               [2] IMPLICIT    NumberDigits,
   explicitPublic               [3]             PublicTON }
```

Type PublicTON is as follows (public type of numbers are derived from CCITT E.164):

```
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 }
```

2.2.24 Reroute Authorization

In Route Select service, this allows the application to specify the chosen destination states which have to set up the ReRoute Request. The latter includes this private data, but only with the state which causes the service.

Note:

This private data is available since R4.1.1 for Routing Services but only “setBusy” state is taken into account.

Since R4.2, when used with RSI (Routing Services Interface), all the states are taken into account.

2.2.24.1 ASN.1 encoding

```
Reroute Authorization          OBJECT-TYPE
        SYNTAX                 RerouteAuthorization
        ACCESS                 read-write
        STATUS                 mandatory
        ::={ 1 3 0012 1206 524 }

   -- RerouteAthorization description --
RerouteAuthorization            ::= BIT STRING
{   Busy                       [0]
    DestNotObtainable          [1]
    IncompatibleDesti          [2]
    NetworkCongestion          [3]
RessourceNotAvailable          [4]
TrunkBusy                      [5]
OtherCase                      [255] }
```

2.2.25 Additional Digits Reporting Criteria

In the Start Data Collection escape service, il allows additional parameters to be added.

2.2.25.1 ASN.1 encoding

```
digitsReportingCriteriaAdd      OBJECT-TYPE
      SYNTAX                    DigitsReportingCriteriaAdd
      ACCESS                    read-write
      STATUS                    mandatory
      ::={ 1 3 0012 1206 525 }

   DigitsKey                    ::=IA5String(SIZE(1..2))

DigitsReportingCriteriaAdd       ::= SEQUENCE
{  abortDigits                  [0]IMPLICIT DigitKey      OPTIONAL,
   ignorDigits                  [1] IMPLICIT DigitKey     OPTIONAL,
   backspaceDigits              [2] IMPLICIT DigitKey     OPTIONAL,
   termDigits                   [3] IMPLICIT DigitKey     OPTIONAL,
   resetDigits                  [4] IMPLICIT DigitKey     OPTIONAL,
   startTimeout                 [5] IMPLICIT INTEGER      OPTIONAL,
   digitTimout                  [6] IMPLICIT INTEGER      OPTIONAL, }
```

2.2.26 Announcement or Music

In the Generate Telephony Tones escape service, it allows additional parameters to be added.

2.2.26.1 ASN.1 encoding

```
AnnouncementOrMusic             OBJECT-TYPE
      SYNTAX                    AnnouncementOrMusi
      ACCESS                    read-write
      STATUS                    mandatory
      ::={ 1 3 0012 1206 526 }

   AnnouncementOrMusic          ::=CHOICE
{  announcement                 AnnouncementToGenerate,
   musicId                      INTEGER }

AnnouncementToGenerate          ::= SEQUENCE
{  language                     INTEGER,
   expectedWaitingTime          [0] INTEGER      OPTIONAL,
   positionInQueue              [1] INTEGER      OPTIONAL,
   SEQUENCE OF CHOICE
   {  digitsList                [0] DigitsToPlay,
      prompt                    [1] Prompt}
Prompt                          ::= SEQUENCE
{  interruptible                BOOLEAN,
   id  INTEGER }
SpellingOption                  ::= ENUMERATED
{  numberByNumber               (0),
   date                         (1)
   hour                         (2)
   phoneNumber                  (3)
   currency                     (4) }
DigitsToPlay                    ::= SEQUENCE
{  spellingOption               [0] SpellingOption,
   digits                       [1] IA5String (SIZE(1..255)) }
```

2.2.27 Global Call ID List

Used in the snapshot device result to report the GlobalCallID linked with each Call ID existing in that service

2.2.27.1 ASN.1 encoding

```
globalCallIDList                OBJECT-TYPE
    SYNTAX                      GlobalCallIDList
    ACCESS                      read-write
    STATUS                      mandatory
    ::={ 1 3 0012 1206 527 }

  globalCallIDList              ::=SEQUENCE OF globalCallID
```

2.2.28 Local Hybrid Link Flag

2.2.28.1 Description

This appears in the Network Reached, Delivered, Queued Established, Transferred, Conferenced events and in Route Request service when a call is done by a local hybrid link

2.2.28.2 ASN.1 encoding

```
localHybridLinkFlag          OBJECT-TYPE
    SYNTAX                   LocalHybridLinkFlag
    ACCESS                   read-write
    STATUS                   mandatory
    ::={ 1 3 0012 1206 528 }

  -- Alcatel-Lucent Hybrid Link Flag description --
LocalHybridLinkFlag ::= BOOLEAN
```

2.2.29 DeviceType Information

2.2.29.1 Description

This is used in the Loggon event, Agent Logging Information escape service and Query Device function result to indicate the type of device (monoline, multiline, agent, supervisor, pilot,..;)

2.2.29.2 ASN.1 encoding

```
deviceTypeInformation          OBJECT-TYPE
      SYNTAX                   DeviceTypeInformation
      ACCESS                   read-write
      STATUS                   mandatory
      ::={ 1 3 0012 1206 529 }
   --Alcatel-Lucent Device Type Information description—
AcdStationDeviceType::= ENUMERATED
{  standard                    (0),
   proacd                      (1),
   agent                       (2),
   supervisor                  (3),
   svi(4) }
StationDeviceType::= SEQUENCE
{  multiline                   BOOLEAN
   acdStationType              AcdStationDeviceType }
AcdDeviceType::= ENUMERATED
{  pilot                       (0),
   rsi (1) }
DeviceTypeInformation ::= CHOICE
{  stationType                 [0] IMPLICIT    StationDeviceType,
   acdType                     [0] IMPLICIT    StationDeviceType }
```

2.2.30 DeviceType Information

2.2.30.1 Description

This appears in the Delivered, Queued Established, Transferred, Conferenced events when a call is an ACD or RSI call.

2.2.30.2 ASN.1 encoding

```
acdCallInformation          OBJECT-TYPE
    SYNTAX                  AcdCallInformation
    ACCESS                   read-write
    STATUS                   mandatory
    ::={ 1 3 0012 1206 530 }

  --Alcatel-Lucent Device Type Information description—

AcdCallInformationInformation ::= CHOICE
{  pilotNumber          [0]          DeviceID,
   rsiNumber            [0]          DeviceID }
```

2.2.31 Other Private Data

2.2.31.1 Description

Allows the definition of non Alcatel-Lucent private data.

2.2.31.2 ASN.1 encoding

```
otherPrivateData                OBJECT-TYPE
        SYNTAX                  OtherPrivateData
        ACCESS                  read-write
        STATUS                  mandatory
        ::= { 1 3 0012 1206 999 }

OtherPrivateData::=             SEQUENCE
{   private Identifier          [0] IMPLICIT OCTET STRING,
    privateData                 [1] IMPLICIT OCTET STRING    OPTIONAL }
```

2.2.31.3 Fields description

- *privateIdentifier:* identifier for this private data
- *privateData*: data bytesP

2.3 Private events

These events have the object identifier range 1.3.0012.1206.1000 to 1.3.0012.1206.1499 .

2.3.1 Head Of Queue

2.3.1.1 Description

With this information, the IVR is able to shodcut the non interruptible transaction, assuming it is possible. The CCD distribution indicates to the IVR that the call has reached the head of the queue. This event will be sent only once.

2.3.1.2 ASN.1 encoding

```
headOfQueueEvent            OBJECT-TYPE
    SYNTAX                   HeadOfQueueEvent
    ACCESS                  read-write
    STATUS                   mandatory
    ::= { 1 3 0012 1206 1000 }

HeadOfQueueEvent ::= ConnectionID
```

Note:

This event was previously defined for "Internal use".

2.3.2 Supervisor Assist Request

2.3.2.1 Description

This event is relative to the private operations Supervisor Assist Request/Cancel (see explanations in sections 2.1.7 and 2.1.8).

2.3.2.2 ASN.1 encoding

```
supervisorAssistRequestEvent      OBJECT-TYPE
        SYNTAX                    SupervisorAssistRequestEvent
        ACCESS                    read-write
        STATUS                    mandatory
        ::= { 1 3 0012 1206 1001 }
SupervisorAssistRequestEvent      ::= SEQUENCE
{   agentDevice                   DeviceID,
    agentConnection               ConnectionID,
    supervisorDevice              DeviceID,
    requestStatus                 BOOLEAN
    correlatorData                CorrelatorData      OPTIONAL }
```

2.3.2.3 Fields description

- *agentDevice:* agent number requesting the support of a supervisor
- *agentConnection:* agent connection identifier
- *supervisorDevice:* supervisor number
- *requestStatus:* TRUE means a Supervisor Assist Request operation; FALSE indicates a Supervisor Assist Cancel operation

2.3.3 Trunk Group Supervision

This private even t was developed following an operator request and is available from AEnterprise release R2.1.

2.3.3.1 ASN.1 encoding

```
trunkGroupSupervisionEvent      OBJECT-TYPE
      SYNTAX                    TrunkGroupSupervisionEvent
      ACCESS                    read-write
      STATUS                    mandatory
   ::= { 1 3 0012 1206 1002 }
TrunkGroupSupervisionEven       t ::= SEQUENCE
{  trunkGroupNumberList                          OCTET STRING,
   outOfServiceTrunkGroupNumberList              OCTET STRING,
   busyTrunkGroupNumberList                      OCTET STRING,
   freeTrunkGroupNumberList                      OCTET STRING }
```

2.3.4 Roaming DECT

2.3.4.1 Description

Roaming DECT localisation is a switch feature allowing a DECT subscriber to move transparently from an OmniPCX Enterprise node to another. In the first step, it does not provide a single directory number for monitoring the handset independently of the OmniPCX Enterprise node into which it is located.

This private event will be then generated if a DECT set moves from one OmniPCX Enterprise node to an other one.

2.3.4.2 ASN.1 encoding

```
MovingState                    ::= ENUMERATED
{ to-another-node (0), to-home-node(1) }
roamingDectEvent               OBJECT-TYPE
      SYNTAX                   RoamingDectEvent
      ACCESS                   read-write
      STATUS                   mandatory
      ::= { 1 3 0012 1206 1003 }
RoamingDectEvent               ::= SEQUENCE
{  moving                      MovingState,
   newNode                     [10] IMPLICIT INTEGER   OPTIONAL,
   subNetworkNb                [11] IMPLICIT INTEGER   OPTIONAL,
   shellNumber                 DeviceID                OPTIONAL }
```

2.3.4.3 Fields description

moving: DECT set has moved to an Cher node or is returned toits home node (the OmniPCX Enterprise node on which the DECT set has been registered); if *moving* is set with *to-another-no*de, fields *newNode, subNetworkNb* and *shellNumber* will be filled (NULL values otherwise)

- *newNode:* new destination node
- *subNetworkNb:* sub-network where the set is
- *shellNumber:* shell directory number of the device on the new node

Note:

If the set moves to a new node, the monitoring request of this set (on the new node) should be done on the shell number.

2.3.5 Telephony TonesGenerated Event

This event indicates that telephony tones have been generated or stopped at a device. It is relative to the private operations Generate Telephony Tones and CancelTelephony Tones for the RSI feature.

2.3.5.1 ASN.1 encoding

```
telephonyTonesGenerateEvent     OBJECT-TYPE
        SYNTAX                  TelephonyTonesGenerateEvent
        ACCESS                  read-write
        STATUS                  mandatory
        ::= { 1 3 0012 1206 1004 }
OtherPrivateEvent ::=           SEQUENCE
{   monitorCrossRefID           monitorCrossRefID3
    connection                  ConnectionID3
    toneGenerated               TelephoneTone3          OPTIONAL
    toneFrequency               [0] IMPLICIT INTEGER    OPTIONAL,
    toneDuration                [1] IMPLICIT INTEGER    OPTIONAL,
    pausedDuration              [2] IMPLICIT INTEGER    OPTIONAL,
    connectionInfo              ConnectionInformation3  OPTIONAL,
    extensions                  CSTACommonArguments3    OPTIONAL }
```

2.3.6 Busy Event

This private event indicates that a call could not reach a CSTA monitored device due to a "totally busy", "do not disturb" state or forwarding overflow…without prompting configured for this device (the reason is indicated in the cause field).

2.3.6.1 ASN.1 encoding

```
BusyEvent                       OBJECT-TYPE
      SYNTAX                    BusyEvent
      ACCESS                    read-write
      STATUS                    mandatory
      ::= { 1 3 0012 1206 1005 }
BusyEvent                       ::= SEQUENCE
{   failingDevice               SubjectDeviceID
    callingDevice               CallingDeviceID
    calledDevice                CalledDeviceID,
    cause                       EventCause }
```

2.3.7 Remote Record Failed Event

This private event indicates that a DR-Link (Voice Logger) listening operation in network could not be performed due to network problems (the reason is indicated in the cause field).

2.3.7.1 ASN.1 encoding

```
remoteRecordFailedEvent          OBJECT-TYPE
      SYNTAX                     RemoteRecordFailedEvent
      ACCESS                     read-write
      STATUS                     mandatory
      ::= { 1 3 0012 1206 1006 }
RemoteRecordFailedEvent          ::= SEQUENCE
{  recordedDevice                SubjectDeviceID
   connection                    ConnectionID
   globalCallIDe                 GlobalCallID,
   cause                         EventCause }
```

2.3.8 Other Private Event

2.3.8.1 Description

Allows to define non Alcatel-Lucent private events.

2.3.8.2 ASN.1 encoding

```
otherPrivateEvent                OBJECT-TYPE
    SYNTAX OtherPrivateEvent ACCESS read-write
    STATUS mandatory
    ::= { 1 3 0012 1206 1499 }
OtherPrivateEvent ::=            SEQUENCE
{   privateEventIdentifier       [0] IMPLICIT OCTET STRING,
    privateEvent                 [1 ] IMPLICIT OCTET STRING    OPTIONAL }
```

2.3.8.3 Fields description

- *privateEventidentifier* identifier for this private event
- *privateEvent:* data bytes

Chapter 3

Summary

Listed below is the existing Alcatel-Lucent private information and the corresponding CSTA version where it is available:

Alcatel-Lucent Private Information	ECMA CSTA Phase I	ECMA CSTA Phase II	ECMA CSTA Phase III	OmniPCX Enterprise Release
Private operations				
Associate Data 1.3.12.1206.0	Yes^{O}	No (1)		All
Set Date And Time 1.3.12.1206.1	Yes	Yes		up to R3.2
Host Information 1.3.12.1206.2	No	Yes		All
Interrupt Transaction 1.3.12.1206.3	No	Yes		R2.0
Set Device In Service 1.3.12.1206.4	No	Yes		All
Fast Data 1.3.12.1206.5	Yes	No (1)		R1.4
Supervisor Assist Request 1.3.12.1206.6	No	Yes		R3.0
Supervisor Assist Cancel 1.3.12.1206.7	No	Yes		R3.0
Escape Register Request 1.3.12.1206.8	No	Yes		R2.1
EscapeRegister Cance l 1.3.12.1206.9	No	Yes		R2.1
Agent Logging Information 1.3.12.1206.10	No	Yes		R2.1

(1) provided by the CSTA version 2 protocol

Alcatel-Lucent Private Information	ECMA CSTA Phase I	ECMA CSTA Phase II	OmniPCX Enterprise Release

Private operations (continued)			
Start Listening 1.3.12.1206.11	No	Yes	R3.1
Stop Listening 1.3.12.1206.12	No	Yes	R3.1
Beep Tone Generation 1.3.12.1206.13	No	Yes	R3.1
Roaming DECT Status 1.3.12.1206.14	No	Yes	R3.1
Permanent Listening Acivation 1.3.12.1206.15	No	Yes	R4.1
Call CenterTreatment Request 1.3.12.1206.16	No	Yes	R4.1
Set Message Waiting Indicator 1.3.12.1206.17	No	Yes	R4.1
Start Data Collection 1.3.12.1206.18	No	Yes	R4.2
Stop Data Collection 1.3.12.1206.19	No	Yes	R4.2
Stop Data Collection 1.3.12.1206.20	No	Yes	R4.2
Stop Data Collection 1.3.12.1206.20	No	Yes	R4.2
Generate Telephony Tones 1.3.12.1206.21	No	Yes	R4.2
Cancel Telephony Tones 1.3.12.1206.22	No	Yes	R4.2
Other Operation 1.3.12.1206.499	Yes	Yes	All

Alcatel-Lucent Private Information	ECMA CSTA Phase I	ECMA CSTA Phase II	OmniPCX Enterprise Release
Private data			
Correlator Data 1.3.12.1206.500	Yes	No (1)	All
Private Errors 1.3.12.1206.501	Yes	No (1)	All
CCD Pilot Expected Waiting Time 1.3.12.1206.502	No	Yes	All
Service Option 1.3.12.1206.503	Yes	Yes	All (2)
Network Time Slot 1.3.12.1206.504	Yes	Yes	All (5)
Not Ready Context 1.3.12.1206.505	No	Yes	R2.0
CCD Agent Assignment Information 1.3.12.1206.506	No	Yes	All
CCD Pilot Status 1.3.12.1206.507	No	Yes	All (3)
CCD Delivered Call Information 1.3.12.1206.508	No	Yes	All
Interactive Queuing Delivered Call Info 1.3.12.1206.509	No	Yes	R2.0
Interactive Queuing Reconnect Guide Level 1.3.12.1206.510	No	Yes	R2.0
Withdrawal Type 1.3.12.1206.511	No	Yes	R2.0
Party Name 1.3.12.1206.512	No	Yes	R2.1
Requesting Device 1.3.12.1206.513	No	Yes	R2.1 (6)
Rerouted Call Indication 1.3.12.1206.514	No	Yes	R2.1 (4)
Host Identification 1.3.12.1206.515	No	Yes	R2.1

Alcatel-Lucent Private Information	ECMA CSTA Phase I	ECMA CSTA Phase II	OmniPCX Enterprise Release
Private data (continued)			
Supervised Transfer 1.3.12.1206.516	No	Yes	R3.1
reserved 1.3.12.1206.517	No	Yes	R4.1
ACR Attributes List 1.3.12.1206.518	No	Yes	R4.1
CCD Treatment Type 1.3.12.1206.519	No	Yes	R4.1
Secrecy Identity 1.3.12.1206.520	No	Yes	R4.1
Global Call ID 1.3.12.1206.521	No	Yes	R4.1.1
Old Global Call ID 1.3.12.1206.522	No	Yes	R4.1.1
Nationa/International Indicator 1.3.12.1206.523	No	Yes	R4.1
Reroute Authorization 1.3.12.1206.524	No	Yes	R4.1.1
Digits Reporting Criteria Add 1.3.12.1206.525	No	Yes	R4.2
Announcement or Music 1.3.12.1206.526	No	Yes	R4.2
Global Call ID List 1.3.12.1206.527	No	Yes	R4.2
Local Hybrid Link Flag 1.3.12.1206.528	No	Yes	R5.0.1Ux
Device Type Information 1.3.12.1206.529	No	Yes	R6.0
Acd Call Information 1.3.12.1206.530	No	Yes	R6.0
Other Private Data 1.3.12.1206.999	Yes	Yes	All

(1) provided by the CSTA version 2 protocol

(2) option *autoOriginate* is defined in OmniPCX Enterprise release R2.1
supervisorCall is defined in OmniPCX Enterprise release R3.0
supervisorStepIn is defined in OmniPCX Enterprise release R3.0
supervisedTransfer is defined in OmniPCX Enterprise release R3.0
headsetMode is defined in OmniPCX Enterprise release R3.0
backupRouting is defined in OmniPCX Enterprise release R4.2

(3) in OmniPCX Enterprise B3.35x, the CCD Pilot Status is only in the Query Device result to a Pilot number

(4) information migrated from specific private data to generic private data

(5) reports in events Transferred and Conferenced is defined in OmniPCX Enterprise R2.1

(6) Alcatel-Lucent 4980 Nomadic support is defined in OmniPCX Enterprise R3.1

Alcatel-Lucent Private Information	ECMA CSTA Phase I	ECMA CSTA Phase II	OmniPCX Enterprise Release
Private events			
Head Of Queue Event 1.3.12.1206.1000	No	Yes	R2.0 (5)
Supervisor Assist Request 1.3.12.1206.1001	No	Yes	R3.0
Trunk Group Supervision Event 1.3.12.1206.1002	No	Yes	R2.1
Roaming DECT Event 1.3.12.1206.1003	No	Yes	R3.1
Telephony Tones Generated Event 1.3.12.1206.1004	No	Yes	R4.2
Busy Event 1.3.12.1206.1005	No	Yes	R4.2
Remote record Failed Event 1.3.12.1206.1006	No	Yes	R4.2
Other Private Event 1.3.12.1206.1499	Yes	Yes	All

3.1 Private data in CSTA events and CSTA services

The following table indicates the private data used in the CSTA events:

CSTA event	Private data
Conferenced	Network Time Slot Universal Call Identifier Reminder (from OmniPCX R4.1.1)
Delivered	CCD Pilot Expected Waiting Time CCD Pilot Status CCD Delivered Call Information Interactive Queueing Delivered Call Information Network Time Slot Party Name Rerouted Call Information Supervised Transfer CCD Treatment Type (from OmniPCX R4.1.1) National/International Indicator
Established	CCD Delivered Call InformationNational/International Indicator
Logged On/Off	Party NameRequesting Device
Not Ready	Withdrawal Type
Queued	Party Name Network Time Slot National/International Indicator
Ready	CCD Treatment Type (from OmniPCX R4.1.1)
Transferred	Network Time Slot Universal Call Identifier Reminder (from OmniPCX R4.1.1)
Working After Call	Network Time Slot CCD Treatment Type

Note:

Data Secrecy Identity and Universal Call Identifier are available for all call control events.

The following table indicates the private data used in the CSTA services :

CSTA service	Private data
Clear Connection	Interactive Queueing Reconnect Guide Level Requesting Device
Consultation Call	Service Option callProgressToneInhibition Service Option holdToneInhibition Service Option prioritaryTransfer Service Option supervisorCall Service Option supervisedTransfer Supervised Transfer
Make Call	Service Option callProgressToneInhibition Service Option supervisorCall Service Option autoOriginate Requesting Device
Query Agent State result	CCD Agent Assignment Information Requesting Device
Query Device Information request to a Pilot number	Service Option prioritaryTransfer
Query Device Information result from a Pilot number	CCD Pilot Status CCD Pilot Expected Waiting Time Supervised Transfer
Set Feature Logged On	Service Option headsetMode
Set Feature Not Ready	Withdrawal Type
Single Step Conference	Service Option supervisorStepIn
System Status	Host Identification

Note:

Universal Call Identifier is provided in acknowledgement of following CSTA services: Make Call, Consultation Call, Transfer Call and Conference Call

Acronyms and Definitions

Acronyms

ABC–F	Alcatel-Lucent Business Communication
ACD	Automatic Call Distribution
ACSE	Association Control Service Element
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
BUSY	Busy tone
CAS	Channel Associating Signaling
CCD	Call Centre Distribution
CCS	Common Channel Signaling
CSTA	Computer Supported Telecommunications Applications
CTI	Computer and Telephony Integration
DASS2	Digital Access Signaling System 2
DPNSS	Digit al Private Network Signaling System
DSP	Digital Signal Processor
DTO	Tone Detection
ECMA	European Computer Manufacturer Association
GAP	Generic Access Profile
GPA	General Purpose Auxiliary board
GSM	Global System Mobile
I/O	Input/Output
ISO	International Organisation for Standardisation

ISDN	Integrated Services Digital Network
IVR	Interactive Voice Response
LE	Low End
MAO	Management Objects
NDD	None Direct Dialing In
PCX	Private branch exchange
PCM	Pulse Code Modulation
QSIG	Q Interface Signaling
RINGING	Ringing tone
TSA	Telephony Server Assistant
Tx	filter on the detection of voice signal
Ty	filter on the non detection of voice signal
UA	Universal Alcatel-Lucent
VAD	Voice Activity Detection
VVLE	Very Very Low End

Definitions

Agent	a CSTA user associated with one or more ACD devices or ACD groups and authorized to act on behalf of the provider of the CSTA application
B channel	a 56 or 64 Kbps channel on an ISDN or proprietary PBX line that can carry voice or data
Computing Function	the part of the domain needed to support CSTA applications that is also within a Computing or Special Resource sub– domain
Connection identifier	an identifier used to identify a relationship between a specific call and a specific device; the Connection Identifier comprises a Call Identifier and a Device Identifier; together, these identifiers specify a unique CSTA Object in the context of a CSTA Association
Correlator Data	computing domain–specific data associated with a call and used to track a call as it is controlled and monitored by the computing function
CSTA events	a message provided by the switching function to the computing function to indicate a change of the state of a CSTA object; events are subcategorized into Call Control, Call Associated, Media Stream, Physical Device, Logical Device, Media Attachment, Voice Unit, Maintenance, and Private events
I/O service	service allowing a computing function to send a data stream to or receive a data stream from a device in a switching sub– domain
Monitoring services	the services provided by the switching function by which the computing function may receive notification of changes in the switching function; the computing function indicates interest in certain switching function changes, and thereafter receives notifications of those changes via events
Private information	implementation–specific information
Status Reporting service	service through which system and other status information is passed between the Switching Function and the Computing Function
Switching Function	the part of the domain needed to support CSTA applications that is implemented within a switching sub–domain

Switching Function service	a service provided by the switch that can be invoked by a computing function or by manual telephone activity
Voice Unit	a special resource function that allows messages consisting of voice stream data to be created, manipulated, played to a connection, or recorded to a connection