Standard ECMA-314
2nd Edition - December 2001

ECMA

Standardizing Information and Communication Systems

Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Identification and Call Linkage Additional Network Feature

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Standard ECMA-314
2nd Edition - December 2001

ECMA

Standardizing Information and Communication Systems

Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Identification and Call Linkage Additional Network Feature

(QSIG-CIDL)

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Brief History

This Standard is one of a series of ECMA Standards defining services and signalling protocols applicable to Private Integrated Services Networks (PISNs). The series uses ISDN concepts as developed by ITU-T and conforms to the framework of International Standards for Open Systems Interconnection as defined by ISO/IEC.

This particular Standard specifies the signalling protocol for use at the Q reference point in support of the Call Identification and Call Linkage Additional Network Feature. The protocol defined in this Standard forms part of the PSS1 protocol (informally known as QSIG).

This Standard is based upon the practical experience of ECMA member companies and the results of their active and continuous participation in the work of ISO/IEC JTC1, ITU-T, ETSI and other international and national standardization bodies. It represents a pragmatic and widely based consensus.

Compared to the 1st Edition of Standard ECMA-314 (published by ECMA in September 2000), this 2nd Edition incorporates migration to ASN.1 version 1997 as well as changes in order to achieve complete alignment with International Standard ISO/IEC 13870:2001(E) published by ISO/IEC in 2001.

Adopted as 2nd Edition of Standard ECMA-314 by the General Assembly of December 2001.

Table of contents

1 Scope

This Standard specifies the signalling protocol for the support of the Call Identification and Call Linkage additional network feature (ANF-CIDL) at the Q reference point between Private Integrated Network services eXchanges (PINXs) connected together within a Private Integrated Services Network (PISN).

ANF-CIDL is an additional network feature which allows the assignment of a Global Call Identification (GID) to identify a call end-to-end over the call route (i.e. between the two end PINXs). As an option, a Thread Identification (TID) may be assigned to different calls which are logically linked together due to the operation of other supplementary services and/or ANFs. Additionally a Leg Identification (LID) may be assigned, to identify the different legs of a call.

NOTE 1
This ANF has been developed to support the use of CSTA (ECMA-269) in a networked environment, i.e. in a PISN. Use of this ANF for other applications is not precluded.

The Q reference point is defined in ECMA-133.

Supplementary Service specifications are produced in three stages and according to the method specified in ETS 300 387. This Standard contains the stage 3 specification for the Q reference point and satisfies the requirements identified by the stage 1 and stage 2 specifications in ECMA-313.

The signalling protocol for ANF-CIDL operates on top of the signalling protocol for basic circuit switched call control, as specified in ECMA-143, and uses certain aspects of the generic procedures for the control of supplementary services specified in ECMA-165.

This Standard also specifies additional signalling protocol requirements for the support of interactions at the Q reference point between Call Identification and Call Linkage and other supplementary services and ANFs.

This Standard is applicable to PINXs which can be interconnected to form a PISN.

2 Conformance

In order to conform to this Standard, a PINX shall satisfy the requirements identified in the Protocol Implementation Conformance Statement (PICS) proforma in annex A.

Conformance to this Standard includes conforming to those clauses that specify protocol interactions between ANF-CIDL and other supplementary services and ANFs for which signalling protocols at the Q reference point are supported in accordance with the stage 3 standards concerned.

3 References (normative)

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

In the case of references to ECMA Standards that are aligned with ISO/IEC International Standards, the number of the appropriate ISO/IEC International Standard is given in brackets after the ECMA reference.

ECMA-133	Private Integrated Services Network (PISN) - Reference Configuration for PISN Exchanges (PINX) (International Standard ISO/IEC 11579-1)
ECMA-142	Private Integrated Services Network (PISN) - Circuit Mode 64kbit/s Bearer Services - Service Description, Functional Capabilities and Information Flows (International Standard ISO/IEC 11574)
ECMA-143	Private Integrated Services Network (PISN) - Circuit Mode Bearer Services - Inter-Exchange Signalling Procedures and Protocol (International Standard ISO/IEC 11572)
ECMA-155	Private Integrated Services Networks - Addressing (International Standard ISO/IEC 11571)

ECMA-165	Private Integrated Services Network (PISN) - Generic Functional Protocol for the Support of Supplementary Services - Inter-Exchange Signalling Procedures and Protocol (International Standard ISO/IEC 11582)
ECMA-174	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Diversion Supplementary Services (International Standard ISO/IEC 13873)
ECMA-176	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Path Replacement Additional Network Feature (International Standard ISO/IEC 13874)
ECMA-178	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Transfer Supplementary Service (International Standard ISO/IEC 13869)
ECMA-186	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Completion Supplementary Services (International Standard ISO/IEC 13870)
ECMA-192	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Offer Supplementary Service (International Standard ISO/IEC 14843)
ECMA-194	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Do Not Disturb and Do Not Disturb Override Supplementary Services (International Standard ISO/IEC 14844)
ECMA-203	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Intrusion Supplementary Service (International Standard ISO/IEC 14846)
ECMA-221	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Interception Additional Network Feature (International Standard ISO/IEC 15054)
ECMA-264	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Call Priority Interruption and Call Priority Interruption Protection Supplementary Services (International Standard ISO/IEC 15992)
ECMA-269	Services for Computer Supported Telecommunications Applications (CSTA) Phase III (International Standard ISO/IEC 18051)
ECMA-282	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Private User Mobility (PUM) - Registration Supplementary Service (International Standard ISO/IEC 17876)
ECMA-284	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Private User Mobility (PUM) - Call Handling Additional Network Features (International Standard ISO/IEC 17878)
ECMA-300	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Single Step Call Transfer Supplementary Service (International Standard ISO/IEC DIS 19460)
ECMA-302	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Wireless Terminal Location Registration Supplementary Service and Wireless Terminal Information Exchange Additional Network Feature (International Standard ISO/IEC 15429)
ECMA-304	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Wireless Terminal Call Handling Additional Network Features (International Standard ISO/IEC 15431)
ECMA-306	Private Integrated Services Network (PISN) - Inter-Exchange Signalling Protocol - Wireless Terminal Authentication Supplementary Services (International Standard ISO/IEC 15433)
ECMA-313	Private Integrated Services Network (PISN) - Specification, Functional Model and Information Flows - Call Identification and Call Linkage Additional Network Feature
ETS 300 387	Private Telecommunication Network (PTN); Method for the specification of basic and supplementary services (1994)

ITU-T Rec. H.225 Call signalling protocols and media stream packetization for packet-based multimedia communication systems (1999)

ITU-T Rec. I.112 Vocabulary of terms for ISDNs (1993)

ITU-T Rec. I.210 Principles of telecommunication services supported by an ISDN and the means to describe them (1993)

ITU-T Rec. Q.950 Supplementary services protocols, structure and general principles (2000)

ITU-T Rec. Z.100 Specification and description language (1999)

4 Definitions

For the purposes of this Standard, the following definitions apply.

4.1 External definitions

This Standard uses the following terms defined in other documents:

– Application Protocol Data Unit (APDU)	(ECMA-165)
– ANF-CIDL user	(ECMA-313)
– Basic Call	(ECMA-143)
– Basic Service	(ITU-T Rec. I.210)
– Call Identification Data, CIDL data	(ECMA-313)
– Call Independent Signalling Connection	(ECMA-165)
– Complete Number	(ECMA-155)
– Gateway PINX	(ECMA-143)
– Global Call Identification, Global Call ID, GID	(ECMA-313)
– Global Thread Identification, Thread ID, TID	(ECMA-313)
– Interpretation APDU	(ECMA-165)
– Leg Identification	(ECMA-313)
– Network Facility Extension (NFE)	(ECMA-165)
– Originating PINX	(ECMA-165)
– Private Integrated Services Network (PISN)	(ECMA-133)
– Private Integrated services Network eXchange (PINX)	(ECMA-133)
– Signalling	(ITU-T Rec. I.112)
– Supplementary Service	(ITU-T Rec. I.210)
– Supplementary Service Control Entity	(ECMA-165)
– Terminating PINX	(ECMA-165)
– Transit PINX	(ECMA-165)
– User	(ECMA-142)

4.2 Other definitions

4.2.1 CIDL PINX

The PINX receiving initial Global Call ID and optionally Thread ID and Leg ID values from the Originating PINX of a call. Upon supplementary / ANF interactions (e.g. after call transfer), the CIDL PINX may generate a new Global Call ID for the resulting call and may correlate two or more logically linked calls together by updating the Thread ID values of these calls.

NOTE 2
The CIDL PINX may be the Originating or the Terminating PINX of a call, as well as any other PINX invoking Supplementary Services / ANFs for this call, e.g. the SS-CT Transferring PINX.

4.2.2 CIDL Transit PINX

A Transit PINX on the call route between the Originating PINX and a CIDL PINX or between two CIDL PINXs, which informs the local ANF-CIDL user about incoming CIDL Data and passes the data on unchanged to the subsequent PINX.

5 Acronyms

ANF-CIDL	Additional Network Feature Call Identification and Call Linkage
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation no. 1
CIDL	Call Identification and Call Linkage
GID	Global Call Identification
ID	Identification
ISDN	Integrated Services Digital Network
LID	Leg Identification
NFE	Network Facility Extension
PICS	Protocol Implementation Conformance Statement
PINX	Private Integrated services Network eXchange
PISN	Private Integrated Services Network
PNP	Private Numbering Plan
SDL	Specification and Description Language
TID	Thread Identification

6 Signalling protocol for the support of ANF-CIDL

6.1 ANF-CIDL description

NOTE 3
If not otherwise stated, the term call is used in the sense of a Basic Call or a Call Independent Signalling Connection.

ANF-CIDL is an additional network feature which allows the assignment of a Global Call Identification (GID) to identify a call end-to-end over the call route (i.e. between the two end PINXs). As an option, a Thread Identification (TID) may be assigned to different calls which are logically linked together due to the operation of other supplementary services and/or ANFs. Additionally a Leg Identification (LID) may be assigned, to identify the different legs of a call.

This additional network feature is applicable to all basic services defined in ECMA-142.

6.2 ANF-CIDL operational requirements

6.2.1 Provision/Withdrawal

Provision and withdrawal shall be in accordance with 6.2.1 of ECMA-313.

6.2.2 Requirements on an Originating PINX

The basic call procedures specified in ECMA-143 shall be supported.

Generic procedures for the call-related control of supplementary services, as specified in ECMA-165 for an End PINX, shall apply.

6.2.3 Requirements on a CIDL PINX

The basic call procedures specified in ECMA-143 shall be supported.

Generic procedures for the call-related control of supplementary services, as specified in ECMA-165 for an End PINX, shall apply. Additionally generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Originating PINX and a Terminating PINX shall apply.

6.2.4 Requirements on a CIDL Transit PINX

The basic call procedures specified in ECMA-143 shall be supported.

Generic procedures for the call-related control and call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for a Transit PINX, shall apply.

6.3 ANF-CIDL coding requirements

6.3.1 Operations

The operations defined in Abstract Syntax Notation number 1 (ASN.1) in table 1 shall apply. The notation is in accordance with ITU-T Rec. X.680 and X.690. The ITU-T Rec. X.208 and X.209 superseded version is in annex E.

Table 1 - Operations in support of ANF-CIDL

```
Call-Identification-and-Call-Linkage-Operations-asn1-97
            {iso(1) standard (0) pss1-call-identification-and-call-linkage (21889)
            call-identification-and-call-linkage-operations-asn1-97 (1)}

DEFINITIONS EXPLICIT TAGS ::=
BEGIN
IMPORTS
          OPERATION
          FROM Remote-Operations-Information-Objects
          {joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

          EXTENSION, Extension{}
          FROM Manufacturer-specific-service-extension-class-asn1-97
          { iso standard pss1-generic-procedures (11582) msi-class-asn1-97 (11)};

CallIdentification-Operations OPERATION ::= { callIdentificationAssign | callIdentificationUpdate }

callIdentificationAssign      OPERATION  ::= {
                                  ARGUMENT               CallIdentificationAssignArg
                                  RETURN RESULT          FALSE
                                  ALWAYS RESPONDS        FALSE
                                  CODE                   local: 105
                                  }

callIdentificationUpdate      OPERATION  ::= {
                                  ARGUMENT          CallIdentificationUpdateArg
                                  RETURN RESULT          FALSE
                                  ALWAYS RESPONDS        FALSE
                                  CODE                   local: 106
                                  }

CallIdentificationAssignArg   ::=   SEQUENCE {
                                 globalCallID     [0] CallIdentificationData,
                                 threadID         [1] CallIdentificationData  OPTIONAL,
                                 legID            [2] CallIdentificationData  OPTIONAL,
                                 extension        ExtensionType               OPTIONAL
                                 }

CallIdentificationUpdateArg   ::=   SEQUENCE {
                                 globalCallID     [0] CallIdentificationData  OPTIONAL,
                                 threadID         [1] CallIdentificationData  OPTIONAL,
                                 legID            [2] CallIdentificationData  OPTIONAL,
                                 extension        ExtensionType               OPTIONAL
                                 }
```

Table 1 - Operations in support of ANF-CIDL (concluded)

```
CallIdentificationData  ::= SEQUENCE {
-- this structure is according to ECMA-269, 12.2.5 (see annex D)
                              switchingSubDomainName  [0] IMPLICIT    SwitchingSubDomainName OPTIONAL,
                              linkageID               CHOICE {
                                  subDomainID             [1] IMPLICIT    SubDomainID,
                                  globallyUniqueID        [2] IMPLICIT    GloballyUniqueID},
                              timeStamp                   [3] IMPLICIT    TimeStamp OPTIONAL
                          }

SwitchingSubDomainName  ::=   IA5String (SIZE(1..64))

GloballyUniqueID   ::=     OCTET STRING (SIZE(1..16))
-- the GloballyUniqueID shall be coded according to ITU-T Recommendation H.225, section 7.6 (see annex D)

ExtensionType           ::=   CHOICE {
                              extension           [3] Extension{{ExampleExtSet}},
                              sequenceOfExt       [4] IMPLICIT SEQUENCE OF Extension{{ExampleExtSet}}
                                      }

ExampleExtSet EXTENSION ::= {...}

SubDomainID             ::=   OCTET STRING (SIZE(1..8))

TimeStamp               ::=   GeneralizedTime (SIZE(16..19))

END -- of Call-Identification-and-Call-Linkage-Operations-asn1-97
```

6.3.2 Information elements

6.3.2.1 Facility information element

APDUs of the operations defined in 6.3.1 shall be coded in the Facility information element in accordance with ECMA-165.

When conveying the invoke APDUs of the operations defined in 6.3.1, the destinationEntity data element of the NFE shall contain value anyTypeOfPINX, the Interpretation APDU shall contain value discardAnyUnrecognisedInvokePdu.

6.3.2.2 Other information elements

Information elements used during the establishment of the new connection shall be coded as specified in ECMA-143.

6.3.3 Messages

Except for cases where a basic call message is to be conveyed at the same time, the Facility information element shall be conveyed in a FACILITY message as specified in ECMA-165.

6.4 ANF-CIDL state definitions

6.4.1 States at a Originating PINX

The procedures at the Originating PINX are written in terms of the following conceptual states existing within the ANF-CIDL control entity in that PINX in association with a particular call.

6.4.1.1 CIDL-Idle

ANF-CIDL is not operating.

6.4.1.2 CIDL-Simultaneous-Assignment-Supervision

A callIdentificationAssign invoke APDU has been sent and Timer T1 has been started.

6.4.2 States at a CIDL PINX

The procedures at the CIDL PINX are written in terms of the following conceptual states existing within the ANF-CIDL control entity in that PINX.

6.4.2.1 CIDL-Idle

ANF-CIDL is not operating.

6.4.2.2 CIDL-Simultaneous-Assignment-Supervision

A callIdentificationUpdate invoke APDU has been sent or a callIdentificationAssign or a callIdentificationUpdate invoke APDU has been received and Timer T1 has been started.

6.4.3 States at a CIDL Transit PINX

The procedures at the CIDL Transit PINX are written in terms of the following conceptual states existing within the ANF-CIDL control entity in that PINX.

6.4.3.1 CIDL-Idle

ANF-CIDL is not operating.

6.5 ANF-CIDL signalling procedures

References in this clause to protocol control states refer to basic call protocol control states defined in ECMA-143.

NOTE 4
The specification in this sub-clause is based on each of the End PINXs being a different PINX, but this sub-clause is also applicable to scenarios where two or more of the described PINXs are the same. In those scenarios some of the signalling procedures and message flows described in this sub-clause are internal to the PINX implementation and therefore outside the scope of this Standard.

Annex B contains some examples of message sequences.

6.5.1 Actions at a Originating PINX

The SDL representation of procedures at a Originating PINX is shown in C.1 of annex C.

6.5.1.1 Normal procedures

Upon a request for a new call, the Originating PINX

1. shall generate the value of element globalCallID within the CallIdentificationAssignArg anew or, due to supplementary service / ANF interactions, as described in the procedures in 6.8. The content of element globalCallID shall be generated according to ECMA-269, 12.2.5 (see annex D) in a way that its value is unique within the PISN.

2. may additionally generate the value of element threadID within the CallIdentificationAssignArg anew or, due to supplementary service / ANF interactions, as described in the procedures in 6.8. The content of element threadID shall be generated according to ECMA-269, 12.2.5 (see annex D) in a way that its value is unique within the PISN.

3. may additionally generate the element legID within the CallIdentificationAssignArg. The content of element legID shall be generated as described for the globalCallData in ECMA-269, 12.2.5 (see annex D) in a way that its value is unique within the PISN.

4. store the generated values, correlate them to the call and inform the ANF-CIDL user about the generated values.

5. shall send one callIdentificationAssign invoke APDU which shall include all generated elements in the SETUP message of the new call, start timer T1 and enter state CIDL-Simultaneous-Assignment-Supervision.

After entering stage CIDL-Simultaneous-Assignment-Supervision, the Originating PINX shall act as a CIDL PINX for the rest of the call, i.e. it shall perform the actions as described in 6.5.2.

6.5.1.2 Exceptional procedures

Not applicable.

6.5.2 Procedures at the CIDL PINX

The SDL representation of procedures at a CIDL PINX is shown in C.2 of annex C.

6.5.2.1 Normal procedures

In state CIDL-Idle, upon receipt of a callIdentificationAssign invoke APDU in a SETUP message for a call for which no Call Identification Data has been stored, the CIDL PINX.

1. shall store the values received within the elements globalCallID, threadID and legID, correlate them to the call and inform the ANF-CIDL user about the received values.
2. if the CIDL PINX is not an End PINX of the call, it shall send the received Call Identification Data within a callIdentificationAssign invoke APDU towards the End PINX of the call.
3. shall start Timer T1 and enter state CIDL-Simultaneous-Assignment-Supervision.

In state CIDL-Simultaneous-Assignment-Supervision, upon expiry of Timer T1, the CIDL PINX shall enter state CIDL-Idle.

In state CIDL-Idle, upon receipt of a callIdentificationUpdate invoke APDU from another CIDL PINX, the CIDL PINX shall perform the following procedures:

1. if the received callIdentificationUpdate invoke APDU conveyed element legID, the CIDL PINX shall ignore the received legID value.
2. if the received callIdentificationUpdate invoke APDU conveyed element globalCallID, the CIDL PINX
   - shall assign the received globalCallID to the call and
   - if the CIDL PINX is not an End PINX of the call, it shall send the received globalCallID value within a callIdentificationUpdate invoke APDU towards the End PINX of the call;
3. if the received callIdentificationUpdate invoke APDU conveyed element threadID, the CIDL PINX
   - shall assign the received threadID value to the call and
   - if other calls are internally linked to this call (i.e. the same threadID value is assigned to them), the CIDL PINX shall send an internal update request to all of these calls, indicating the received threadID value;
4. if the globalCallID and / or the threadID has been updated due to the received callIdentificationUpdate invoke APDU, the CIDL PINX shall
   - inform the ANF-CIDL user about the new values and
   - start Timer T1 and enter state CIDL-Simultaneous-Assignment-Supervision.
5. if neither the globalCallID nor the threadID has been updated due to the received callIdentificationUpdate invoke APDU, the CIDL PINX shall remain in state CIDL-Idle.

In any state, upon an internal update request for the Global Call ID, due to the actions described in 6.8, the CIDL PINX shall

- generate a new globalCallID,
- inform the ANF-CIDL user about the new value,
- send element globalCallID in a callIdentificationUpdate invoke APDU to the End PINX(s) of the call,
- start Timer T1 and enter state CIDL-Simultaneous-Assignment-Supervision.

In any state, upon an internal update request for the Thread ID, due to the actions described in this sub-clause or 6.8, the CIDL PINX shall

- inform the ANF-CIDL user about the new value,
- send the indicated Thread ID value in element threadID in a callIdentificationUpdate invoke APDU to the End PINX(s) of the call,
- start Timer T1 and enter state CIDL-Assignment-Supervision.

Whenever more then one of the Call Identification data values have to be updated between two CIDL PINXs, the PINX which is updating shall send these values within one callIdentificationUpdate invoke APDU.

When sending a callIdentificationUpdate invoke APDU, those elements which values were updated shall be included, and in addition, also the elements which values were not updated (i.e. which remain unchanged) may be included.

6.5.2.2 Exceptional Procedures

In state CIDL-Simultaneous-Assignment-Supervision, upon receipt of a callIdentificationUpdate invoke APDU from another CIDL PINX, the CIDL PINX shall

1. check whether the received globalCallID value is identical to that internally stored:
   - ⇒ if the values are identical, it shall discard the received globalCallID value.
   - ⇒ if the values are not identical, it shall determine the value to be assigned in the following way:
     - if the globallyUniqueID is used, the globalCallID with the higher binary value of the globallyUniqueID shall be assigned to the call;
     - if the subDomainID is used, the globalCallID with the higher binary value of element subDomainID value shall be assigned to the call;
     - if the internally stored and the received binary values of element globallyUniqueID or subDomainID Value are identical, the globalCallID with the later timeStamp value shall be assigned to the call – if one of the two values does not have a timeStamp value assigned, the one with a timeStamp value shall take precedence;
     - if also the stored and the received timeStamp elements are identical (or not assigned), the globalCallID with the higher binary value of element switchingSubDomainName shall be assigned to the call;
   - ⇒ if the value of the received globalCallID is smaller then the internally stored value, the CIDL PINX shall send a callIdentificationUpdate invoke APDU back on the call reference, on which the callIdentificationUpdate invoke APDU has been received, indicating the unchanged stored values;
   - ⇒ if the values of the received globalCallID are higher then the internally stored values, the CIDL PINX shall:
     - if it is not an End PINX of the call, send the received globalCallID value within a callIdentificationUpdate invoke APDU towards the End PINX of the call and
     - inform the ANF-CIDL user about the new value;
2. check whether the received threadID value is identical to that internally stored:
   - ⇒ if the values are identical, it shall discard the received threadID value;
   - ⇒ if the values are not identical, it shall determine the value to be assigned in the following way:
     - if the globallyUniqueID is used, the threadID with the higher binary value of the globallyUniqueID shall be assigned to the call;

- if the subDomainID is used, the threadID with the higher binary value of element subDomainID Value shall be assigned to the call;
- if the internally stored and the received binary values of element globallyUniqueID or subDomainID Value are identical, the threadID with the later timeStamp value shall be assigned to the call;
- if also the stored and the received timeStamp elements are identical, the threadID with the higher binary value of element switchingSubDomainName shall be assigned;

⇒ if the value of the internally stored threadID is retained, the CIDL PINX shall send a callIdentificationUpdate invoke APDU back on the call reference, on which the callIdentificationUpdate invoke APDU has been received, indicating the unchanged stored value;

⇒ if the value of the received threadID is assigned to the call and if other calls or call segments are internally linked to this call (i.e. the same threadID value is assigned to them), the CIDL PINX shall send an internal update request to all of these calls, indicating the received threadID value and inform the ANF-CIDL user about the new value;

⇒ if both values (i.e. globalCallID and threadID) have to be sent to another PINX due to that process, they shall be sent within one callIdentificationUpdate invoke APDU;

3. restart Timer T1 and remain in state CIDL-Simultaneous-Assignment-Supervision.

In any state, upon receipt of callIdentificationAssign invoke APDU in any other message then a SETUP message, the CIDL PINX shall discard the received values and send a callIdentificationUpdate invoke APDU with the currently stored CIDL Data values to the PINX, which sent the callIdentificationAssign invoke APDU. Afterwards it shall start Timer T1 and enter state CIDL-Simultaneous-Assignment-Supervision.

In any state, upon an internal update request for the Thread ID for a call for which no threadID has been assigned, the CIDL PINX shall act as described in 6.5.2.1 but shall not take that calls into account when determine the threadID for the linked calls according to 6.8. It may send the newly assigned threadID value also to the End PINX of that call.

6.5.3 Actions at a CIDL Transit PINX

The SDL representation of procedures at a CIDL Transit PINX is shown in C.3 of annex C.

6.5.3.1 Normal procedures

Upon receipt of a callIdentificationAssign or callIdentificationUpdate invoke APDU, a CIDL Transit PINX shall forward the therein conveyed Call Identification Data to a local ANF-CIDL user, generate a new callIdentificationAssign or callIdentificationUpdate invoke APDU respectively, containing the Call Identification Data as received and shall send it to the subsequent PINX.

6.5.3.2 Exceptional procedures

Not applicable.

6.6 ANF-CIDL impact of interworking with public ISDNs

As no similar service exists within the public ISDN, interworking is not applicable.

6.7 ANF-CIDL impact of interworking with non-ISDNs

Interworking with similar services in non-ISDNs may be described in an extra ECMA Standard. Therefore it is out of the scope of this Standard.

6.8 Protocol Interactions between ANF-CIDL and other supplementary services and ANFs

This clause specifies protocol interactions with other supplementary services and ANFs for which stage 3 standards had been published at the time of publication of this Standard. For interactions with supplementary services and ANFs for which stage 3 standards are published subsequent to the publication of this Standard, see those other stage 3 standards.

NOTE 5
Simultaneous conveyance of APDUs for ANF-CIDL and another supplementary service or ANF in the same message, each in accordance with the requirements of its respective stage 3 standard, does not, on its own, constitute a protocol interaction.

NOTE 6
Additional interactions that have no impact on the signalling protocol at the Q reference point can be found in the relevant stage 1 specifications.

NOTE 7
Due to PINX internal processes or interworking with other protocols, which offer a different set of services, changes of Call Identification and CIDL data may occur more often then described here.

6.8.1 Calling Name Identification Presentation (SS-CNIP)

No protocol interaction.

6.8.2 Connected Name Identification Presentation (SS-CONP)

No protocol interaction.

6.8.3 Completion of Calls to Busy Subscribers (SS-CCBS)

6.8.3.1 Actions at a Originating PINX

The Originating PINX shall send a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the failed original call, together with the ccbsRequest invoke APDU.

6.8.3.1.1 With path reservation

The Originating PINX shall send, together with the ccPathReserve invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call conveying the ccExecPossible invoke APDU.

6.8.3.1.2 Without path reservation

The Originating PINX shall send, together with the ccRingout invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call conveying the ccbsRequest invoke APDU.

6.8.3.2 Actions at a Terminating PINX

If the connection release method is used, the Terminating PINX shall send, together with the ccExecPossible invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call from the Originating PINX, conveying the ccbsRequest invoke APDU.

6.8.4 Completion of Calls on No Reply (SS-CCNR)

6.8.3 shall apply accordingly for the operations and procedures of SS-CCNR.

6.8.5 Call Forwarding Unconditional (SS-CFU)

The Rerouting PINX shall send to the Diverted-To PINX, together with the divertingLegInformation2 invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the values of the original call and element legID set to a new value.

6.8.6 Call Forwarding Busy (SS-CFB)

6.8.5 shall apply.

6.8.7 Call Forwarding No Reply (SS-CFNR)

6.8.5 shall apply.

6.8.8 Call Deflection (SS-CD)

6.8.5 shall apply.

6.8.9 Call Transfer (SS-CT)

6.8.9.1 Call Transfer by Join

In case

- the call from the Primary PINX is an incoming and the call to the Secondary PINX is an outgoing call or
- the calls from the Primary PINX and from the Secondary PINX are both incoming calls and the primary call was established before the secondary call,
- the calls to the Primary PINX and to the Secondary PINX are both outgoing calls and the primary call was established before the secondary call,

the SS-CT Served User PINX shall send to the Primary and the Secondary PINX identical callIdentificationUpdate invoke APDUs with element globalCallID set to a newly generated unique value and element threadID set to the value assigned to the primary call together with the callTransferActive invoke APDU and / or the callTransferComplete invoke APDU with element callStatus set to “answered”.

In case

- the call to the Primary PINX is an outgoing and the call from the Secondary PINX is an incoming call or
- the calls to the Primary PINX and to the Secondary PINX are both incoming calls and the primary call was established after the secondary call,
- the calls to the Primary PINX and to the Secondary PINX are both outgoing calls and the primary call was established after the secondary call,

the SS-CT Served User PINX shall send to the Primary and the Secondary PINX identical callIdentificationUpdate invoke APDUs with element globalCallID set to newly generated unique value and element threadID set to the value assigned to the secondary call together with the callTransferActive invoke APDU and / or the callTransferComplete invoke APDU with element call status set to “answered”.

6.8.9.2 Call Transfer by Rerouting

In case

- the call from the Primary PINX is an incoming and the call to the Secondary PINX is an outgoing call or
- the calls from the Primary PINX and from the Secondary PINX are both incoming calls and the primary call was established before the secondary call,
- the calls to the Primary PINX and to the Secondary PINX are both outgoing calls and the primary call was established before the secondary call,

the SS-CT Served User PINX shall send identical callIdentificationUpdate invoke APDUs with element globalCallID set to a newly generated unique value and element threadID set to the value assigned to the primary call

- to the SS-CT Primary PINX together with the callTransferInitiate invoke APDU and
- to the SS-CT Secondary PINX together with the callTransferIdentify invoke APDU.

In case

- the call to the Primary PINX is an outgoing and the call from the Secondary PINX is an incoming call or
- the calls to the Primary PINX and to the Secondary PINX are both incoming calls and the primary call was established after the secondary call,
- the calls to the Primary PINX and to the Secondary PINX are both outgoing calls and the primary call was established after the secondary call,

the SS-CT Served User PINX shall send identical callIdentificationUpdate invoke APDUs with element globalCallID set to newly generated unique value and element threadID set to the value assigned to the secondary call

- to the SS-CT Primary PINX together with the callTransferInitiate invoke APDU and
- to the SS-CT Secondary PINX together with the callTransferIdentify invoke APDU.

The SS-CT Primary PINX shall send the callIdentificationAssign invoke APDU with element threadID set to the value as received from the SS-CT Served User PINX and elements globalCallID and legID set to new values, to the Secondary PINX, together with the callTransferSetup invoke APDU.

6.8.10 Path Replacement (ANF-PR)

The Cooperating PINX shall send on the new connection, together with the pathReplaceSetup invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the values of the old connection and element legID set to a new value.

6.8.11 Advice Of Charge (SS-AOC)

No protocol interaction.

6.8.12 Call Offer (SS-CO)

In case Path Retention is not used, the Originating PINX shall send to the Terminating PINX, together with the coRequest invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the failed original call, if such a call existed.

The Terminating PINX may send a callIdentificationUpdate invoke APDU to the other End PINX of the original call with element threadID set to the value of the call from the Originating PINX.

6.8.13 Do Not Disturb (SS-DND)

No protocol interaction.

6.8.14 Do Not Disturb Override (SS-DNDO)

In case Path Retention is not used, the Originating PINX shall send to the Terminating PINX, together with the dndOverride invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call which was cleared due to a do-not-disturb condition, if such a call existed.

6.8.15 Recall (SS-RE)

No protocol interaction.

6.8.16 Call Intrusion (SS-CI)

In case Path Retention is not used, the Originating PINX shall send to the Terminating PINX, together with the ciRequest invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the failed original call, if such a call existed.

If a conference type connection is established between the Served User PINX, the Terminating PINX (serving the wanted user) and the Unwanted User PINX, the Terminating PINX shall send a callIdentificationUpdate invoke APDU to the Served User PINX and the Terminating PINX with identical element globalCallID set to a new value and identical element threadID,

⇒ set to the threadID of the call with the Unwanted User PINX, if this call was an incoming call else

⇒ set to the threadID of the intruding call.

If the Unwanted User PINX is isolated, the Served User PINX (serving the wanted user) may send a callIdentificationUpdate invoke APDU to the Unwanted User PINX with the element threadID set to the value of the intruding call.

6.8.17 Call Interception (ANF-CINT)

The Intercepting PINX shall send to the Intercepted-To PINX, together with the cintLegInformation2 invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the values of the unanswered or unsuccessful call and element legID set to a new value.

6.8.18 Transit Counter (SS-TC)

No protocol interaction.

6.8.19 Route Restriction Class (ANF-RRC)

No protocol interaction.

6.8.20 Authentication of the PISN (SS-WTAN)

No protocol interaction.

6.8.21 Authentication of a WTM user (SS-WTAT)

6.8.21.1 Actions at the Visitor PINX

If a getWtatParam invoke APDU is sent, due to the receipt of a authWtmUser invoke APDU, additionally the Visitor PINX may send a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call conveying the authWtmUser invoke APDU.

6.8.21.2 Actions at the Home PINX

If an authWtmUser invoke APDU is sent, due to the receipt of a wtatParamEnq result APDU, additionally the Visitor PINX may send a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call conveying the wtatParamEnqu result APDU.

6.8.22 Wireless Terminal Location Registration (SS-WTLR)

6.8.22.1 Actions at the Home PINX

If a locDelete invoke APDU is sent to the previous Visitor PINX, due to receipt of a locUpdate invoke APDU from the Visitor PINX, the Home PINX additionally may include a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call on which the locUpdate invoke APDU was received.

6.8.22.2 Actions at the Visitor PINX

If a locUpdate invoke APDU is sent to the Home PINX, following an enquiry call to the Previous Visitor PINX, the Visitor PINX additionally may include a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the enquiry call.

6.8.23 Wireless Terminal Mobility Incoming Call (ANF-WTMI)

6.8.23.1 Actions at the WTMI detect PINX

In addition to the wtmiEnquiry invoke APDU to the Home PINX, the WTMI Detect PINX may send a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the incoming call.

6.8.23.2 Action at the Rerouting PINX

The Rerouting PINX shall send to the Visitor PINX, together with the wtmiInform invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the values of the incoming call and element legID set to a new value.

6.8.24 Wireless Terminal Mobility Outgoing Call (ANF-WTMO)

Within the SETUP message towards the Terminating PINX, the Home PINX shall send a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the value assigned to the call conveying the wtmoCall invoke APDU from the Originating PINX and element legID set to a new value.

6.8.25 Message Waiting Indication (SS-MWI)

No protocol interaction.

6.8.26 Private User Mobility Incoming Call (ANF-PUMI)

6.8.26.1 Actions at the PUMI detect PINX

In addition to the pumiEnquiry invoke APDU to the Home PINX, the PUMI detect PINX may send a callIdentificationAssign invoke APDU with elements globalCallId and legID set to new values and element threadID set to the value of the incoming call.

6.8.26.2 Action at the Rerouting PINX

The Rerouting PINX may send to the Visitor PINX, together with the pumiInform invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the values of the incoming call and element legID set to a new value.

6.8.27 Private User Mobility Outgoing Call (ANF-PUMO)

Within the SETUP message towards the Terminating PINX, the Home PINX may send a callIdentificationAssign invoke APDU with elements globalCallID and threadID set to the value assigned to the call conveying the pumoCall invoke APDU from the Originating PINX and element legID set to a new value.

6.8.28 Common Information (ANF-CMN)

No protocol interaction.

6.8.29 Call Priority Interruption (Protection) (SS-CPI(P))

No protocol interaction.

6.8.30 Private User Mobility - Registration (SS-PUMR)

6.8.30.1 Actions at the Home PINX

If a pumDelReg invoke APDU is sent to the previous Visitor PINX, due to receipt of a pumRegistr invoke APDU from the Visitor PINX, the Home PINX additionally may include a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the call on which the pumRegistr invoke APDU was received.

If a pumDe-reg invoke APDU is received from a Remote PINX, the Home PINX may send, together with a pumDelReg invoke APDU a callIdentificationAssign invoke APDU to the Visitor PINX. The callIdentificationAssign invoke APDU shall include elements globalCallID and legID set to new values and element threadID set to the value of the call on which the pumDe-reg invoke APDU was received.

6.8.30.2 Actions at the Visitor PINX

If a pumRegistr invoke APDU is sent to the Home PINX, following an enquiry call to a Directory PINX, the Visitor PINX additionally may include a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value of the enquiry call.

If a pumRegistr invoke APDU is received from a Remote PINX, the visitor PINX may send, together with a pumRegistr invoke APDU a callIdentificationAssign invoke APDU to the Home PINX. The callIdentificationAssign invoke APDU shall include elements globalCallID and legID set to new values and element threadID set to the value of the call on which the pumRegister invoke APDU was received.

6.8.31 Single Step Call Transfer (SSCT)

The Rerouting PINX shall send to the Transferred-To PINX, together with the ssctSetup invoke APDU, a callIdentificationAssign invoke APDU with elements globalCallID and legID set to new values and element threadID set to the value assigned to the connection on which the ssctInitiate invoke APDU has been received.

At the same time, the Rerouting PINX shall send to the Transferred PINX a callIdentificationUpdate invoke APDU with elements globalCallID and threadID, having the same values as sent to the Transferred-To PINX together with the callTransferComplete invoke APDU.

6.8.32 Simple Dialog (SD)

No protocol interaction.

6.9 ANF-CIDL parameter values (timers)

6.9.1 Timer T1

Timer T1 shall operate at the Originating PINX during state CIDL-Simultaneous-Assignment-Supervision. Its purpose is to protect against simultaneous assignment of a globalCallID or a threadID value to a call.

Timer T1 shall operate at the CIDL PINX during state CIDL-Simultaneous-Assignment-Supervision. Its purpose is to protect against simultaneous assignment of a globalCallID or a threadID value to a call.

Timer T1 shall have a value of 2 seconds.

Annex A
(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this Standard shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The 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's PICS proforma;
- by the user or potential user of an implementation, as a basis for initially checking the possibility of interworking with another implementation. While interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs;
- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Instructions for completing the PICS proforma

A.2.1 General structure of the PICS proforma

The PICS proforma is a fixed format questionnaire divided into subclauses each containing a group of individual items. Each item is identified by an item number, the name of the item (question to be answered), and the reference(s) to the clause(s) that specifies (specify) the item in the main body of this Standard.

The "Status" column indicates whether an item is applicable and if so whether support is mandatory or optional. The following terms are used:

m mandatory (the capability is required for conformance to the protocol);

o optional (the capability is not required for conformance to the protocol, but if the capability is implemented it is required for conformance 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;

x prohibited;

c.<cond> conditional requirement, depending on support for the item or items listed in condition <cond>;

<item>:m simple conditional requirement, the capability being mandatory if item number <item> is supported, otherwise not applicable;

<item>:o simple conditional requirement, the capability being optional if item number <item> is supported, otherwise not applicable;

Answers to the questionnaire items are to be provided either in the "Support" column, by simply marking an answer to indicate restricted choice (Yes) or (No), or in the "Not Applicable" column (N/A).

A.2.2 Additional information

Items of Additional information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended that a large quantity will be supplied, and a PICS can be considered complete without such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional information may be entered next to any answer in the questionnaire, and may be included in items of Exception information.

A.2.3 Exceptional information

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the support column for this. Instead, the supplier is required to write into the support column an x.<i> reference to an item of Exception information, and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this Standard. A possible reason for the situation described above is that a defect in the Standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3 PICS proforma for ANF-CIDL

A.3.1 Implementation identification

Supplier	
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)	

Only the first three items are required for all implementations; other information may be completed as appropriate in meeting requirements for full identification.

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

A.3.2 Protocol summary

Protocol version	1.0
Addenda implemented (if applicable)	
Amendments implemented	
Have any exception items been required (see A.2.3)?	No [ ] Yes [ ] (The answer Yes means that the implementation does not conform to this Standard)

Date of statement	

A.3.3 General

Item	Question/feature	Reference	Status	N/A	Support
A1	Behavior as a Originating PINX for ANF-CIDL		m		Yes [ ]
A2	Behavior as a CIDL PINX for ANF-CIDL		m		Yes [ ]
A3	Behavior as a CIDL Transit PINX for ANF-CIDL		o		Yes [ ] No [ ]
A4	Assignment and update of a Global Call ID		m		Yes [ ]
A5	Assignment and update of a Thread ID		o		Yes [ ] No [ ]
A6	Assignment of a Leg ID		o		Yes [ ] No [ ]

A.3.4 Procedures

Item	Question/feature	Reference	Status	N/A	Support
B1	Support of relevant ECMA-143 and ECMA-165 procedures	6.2	m		Yes [ ]
B2	Signalling procedures at an Originating PINX	6.5.1	m		Yes [ ]
B3	Signalling procedures at a CIDL PINX	6.5.2	m		Yes [ ]
B4	Signalling procedures at a CIDL Transit PINX	6.5.3	A3:m	[ ]	Yes [ ]

A.3.5 Coding

Item	Name of Item	Reference	Status	N/A	Support
C1	Sending of callIdentificationAssign invoke APDU at an Originating PINX	6.3	m		Yes [ ]
C2	Receipt of callIdentificationAssign invoke APDU at a CIDL PINX	6.3	m		Yes [ ]
C3	Sending of callIdentificationAssign invoke APDU at a CIDL PINX, if the CIDL PINX is not the End PINX of the call	6.3	m		Yes [ ]
C4	Sending of callIdentificationUpdate invoke APDU at a CIDL PINX	6.3	m		Yes [ ]
C5	Receipt of callIdentificationUpdate invoke APDU at a CIDL PINX	6.3	m		Yes [ ]
C6	Receipt and Sending of callIdentificationAssign invoke APDU at a CIDL Transit PINX	6.3	B4:m	[ ]	Yes [ ]
C7	Receipt and Sending of callIdentificationUpdate invoke APDU at a CIDL Transit PINX	6.3	B4:m	[ ]	Yes [ ]

A.3.6 Timers

Item	Question/feature	Reference	Status	N/A	Support
D1	Support of Timer T1	6.9.1	m		Yes [ ]

A.3.7 Interactions between ANF-CIDL and SS-CCBS

Item	Question/feature	Reference	Status	N/A	Support
E1	Support of SS-CCBS	ECMA-186	o		Yes [ ] No[ ]
E2	Interactions for ANF-CIDL and SS-CCBS at the Originating PINX for Thread ID	6.8.3.1	c.1	[ ]	Yes [ ]
E3	Interactions for ANF-CIDL and SS-CCBS at the Terminating PINX for Thread ID	6.8.3.2	c.1	[ ]	Yes [ ]

c.1: if (A5 and E1) then mandatory, else N/A

A.3.8 Interactions between ANF-CIDL and SS-CCNR

Item	Question/feature	Reference	Status	N/A	Support
F1	Support of SS-CCNR	ECMA-186	o		Yes [ ] No[ ]
F2	Interactions for ANF-CIDL and SS-CCNR at the Originating PINX for Global Call ID	6.8.4	F1:m	[ ]	Yes [ ]
F3	Interactions for ANF-CIDL and SS-CCNR at the Originating PINX for Thread ID	6.8.4	c.2	[ ]	Yes [ ]
F4	Interactions for ANF-CIDL and SS-CCNR at the Terminating PINX for Global Call ID	6.8.4	F1:m	[ ]	Yes [ ]
F5	Interactions for ANF-CIDL and SS-CCNR at the Terminating PINX for Thread ID	6.8.4	c.2	[ ]	Yes [ ]

c.2: if (A5 and F1) then mandatory, else N/A

A.3.9 Interactions between ANF-CIDL and SS-CFU / SS-CDI

Item	Question/feature	Reference	Status	N/A	Support
G1	Support of SS-CFU / SS-CDI	ECMA-174	o		Yes [ ] No[ ]
G2	Interactions for ANF-CIDL and SS-CFU / SS-CDI at the Rerouting PINX for Global Call ID	6.8.5 6.8.8	G1:m	[ ]	Yes [ ]
G3	Interactions for ANF-CIDL and SS-CFU / SS-CDI at the Rerouting PINX for Thread ID	6.8.5 6.8.8	c.3	[ ]	Yes [ ]

c.3: if (A5 and G1) then mandatory, else N/A

A.3.10 Interactions between ANF-CIDL and SS-CFB

Item	Question/feature	Reference	Status	N/A	Support
H1	Support of SS-CFB	ECMA-174	o		Yes [ ] No[ ]
H2	Interactions for ANF-CIDL and SS-CFB at the Rerouting PINX for Global Call ID	6.8.6	H1:m	[ ]	Yes [ ]
H3	Interactions for ANF-CIDL and SS-CFB at the Rerouting PINX for Thread ID	6.8.6	c.4	[ ]	Yes [ ]

c.4: if (A5 and H1) then mandatory, else N/A

A.3.11 Interactions between ANF-CIDL and SS-CFNR / SS-CDA

Item	Question/feature	Reference	Status	N/A	Support
I1	Support of SS-CFNR / SS-CDA	ECMA-174	o		Yes [ ] No[ ]
I2	Interactions for ANF-CIDL and SS-CFNR / SS-CDA at the Rerouting PINX for Global Call ID	6.8.7 6.8.8	I1:m	[ ]	Yes [ ]
I3	Interactions for ANF-CIDL and SS-CNR / SS-CDA at the Rerouting PINX for Thread ID	6.8.7 6.8.8	c.5	[ ]	Yes [ ]

c.5: if (A5 and I1) then mandatory, else N/A

A.3.12 Interactions between ANF-CIDL and SS-CT

Item	Question/feature	Reference	Status	N/A	Support
J1	Support of SS-CT	ECMA-178	o		Yes [ ] No [ ]
J2	Interactions for ANF-CIDL and SS-CT at the SS-CT Served User PINX for Thread ID	6.8.9	c.6	[ ]	Yes [ ]

c.6: if (A5 and J1) then mandatory, else N/A

A.3.13 Interactions between ANF-CIDL and ANF-PR

Item	Question/feature	Reference	Status	N/A	Support
K1	Support of ANF-PR	ECMA-176	o		Yes [ ] No[ ]
K2	Interactions for ANF-CIDL and ANF-PR for Global Call ID	6.8.10	K1:m	[ ]	Yes [ ]
K3	Interactions for ANF-CIDL and ANF-PR for Thread ID	6.8.10	c.7	[ ]	Yes [ ]

c.7: if (A5 and K1) then mandatory, else N/A

A.3.14 Interactions between ANF-CIDL and SS-CO

Item	Question/feature	Reference	Status	N/A	Support
L1	Support of SS-CO	ECMA-192	o		Yes [ ] No [ ]
L2	Interactions for ANF-CIDL and SS-CO for Thread ID	6.8.12	c.8	[ ]	Yes [ ]

c.8: if (A5 and L1) then mandatory, else N/A

A.3.15 Interactions between ANF-CIDL and SS-DNDO

Item	Question/feature	Reference	Status	N/A	Support
M1	Support of SS-DNDO	ECMA-194	o		Yes [ ] No [ ]
M2	Interactions for ANF-CIDL and SS-DNDO for Thread ID	6.8.14	c.9	[ ]	Yes [ ]

c.9: if (A5 and M1) then mandatory, else N/A

A.3.16 Interactions between ANF-CIDL and SS-CI

Item	Question/feature	Reference	Status	N/A	Support
N1	Support of SS-CI	ECMA-203	o		Yes [ ] No [ ]
N2	Interactions for ANF-CIDL and SSCI for Thread ID	6.8.16	c.10	[ ]	Yes [ ]

c.10: if (A5 and N1) then mandatory, else N/A

A.3.17 Interactions between ANF-CIDL and SS-CINT

Item	Question/feature	Reference	Status	N/A	Support
O1	Support of SS-CINT	ECMA-221	o		Yes [ ] No[ ]
O2	Interactions for ANF-CIDL and SS-CINT at the Intercepting PINX for Global Call ID	6.8.17	O1:m	[ ]	Yes [ ]
O3	Interactions for ANF-CIDL and SS-CINT at the Intercepting PINX for Thread ID	6.8.17	c.11	[ ]	Yes [ ]

c.11: if (A5 and O1) then mandatory, else N/A

A.3.18 Interactions between ANF-CIDL and ANF-WTAT

Item	Question/feature	Reference	Status	N/A	Support
P1	Support of ANF-WTAT	ECMA-306	o		Yes [ ] No[ ]
P2	Interactions for ANF-CIDL and ANF-WTAT at the Visitor PINX for Thread ID	6.8.21.1	c.12	[ ]	Yes [ ] No[ ]
P3	Interactions for ANF-CIDL and ANF-WTAT at the Home PINX for Thread ID	6.8.21.2	c.12	[ ]	Yes [ ] No[ ]

c.12: if (A5 and P1) then optional, else N/A

A.3.19 Interactions between ANF-CIDL and SS-WTLR

Item	Question/feature	Reference	Status	N/A	Support
Q1	Support of ANF-WTLR	ECMA-302	o		Yes [ ] No[ ]
Q2	Interactions for ANF-CIDL and ANF-WTLR at the Home PINX for Thread ID	6.8.22.1	c.13	[ ]	Yes [ ] No[ ]
Q3	Interactions for ANF-CIDL and ANF-WTAT at the Visitor PINX for Thread ID	6.8.21.2	c.13	[ ]	Yes [ ] No[ ]

c.13: if (A5 and Q1) then optional, else N/A

A.3.20 Interactions between ANF-CIDL and ANF-WTMI

Item	Question/feature	Reference	Status	N/A	Support
R1	Support of ANF-WTMI	ECMA-304	o		Yes [ ] No[ ]
R2	Interactions for ANF-CIDL and ANF-WTMI at the WTMI detect PINX for Global Call ID	6.8.23.1	R1:m	[ ]	Yes [ ] No[ ]
R3	Interactions for ANF-CIDL and ANF-WTMI at the WTMI detect PINX for Thread ID	6.8.23.1	c.14	[ ]	Yes [ ] No[ ]
R4	Interactions for ANF-CIDL and ANF-WTMI at the Rerouting PINX for Global Call ID	6.8.23.2	R1:m	[ ]	Yes [ ] No[ ]
R5	Interactions for ANF-CIDL and ANF-WTMI at the Rerouting PINX for Thread ID	6.8.23.2	c.14	[ ]	Yes [ ] No[ ]

c.14: if (A5 and R1) then optional, else N/A

A.3.21 Interactions between ANF-CIDL and ANF-WTMO

Item	Question/feature	Reference	Status	N/A	Support
S1	Support of ANF-WTMO	ECMA-304	o		Yes [ ] No[ ]
S2	Interactions for ANF-CIDL and ANF-WTMO for Global Call ID	6.8.24	S1:m	[ ]	Yes [ ] No[ ]
S3	Interactions for ANF-CIDL and ANF-WTMO for Thread ID	6.8.24	c.15	[ ]	Yes [ ] No[ ]

c.15: if (A5 and S1) then optional, else N/A

A.3.22 Interactions between ANF-CIDL and ANF-PUMI

Item	Question/feature	Reference	Status	N/A	Support
T1	Support of ANF-PUMI	ECMA-284	o		Yes [ ] No[ ]
T2	Interactions for ANF-CIDL and ANF-PUMI at the PUMI detect PINX for Global Call ID	6.8.26.1	T1:m	[ ]	Yes [ ] No[ ]
T3	Interactions for ANF-CIDL and ANF-PUMI at the PUMI detect PINX for Thread ID	6.8.26.1	c.16	[ ]	Yes [ ] No[ ]
T4	Interactions for ANF-CIDL and ANF-PUMI at the Rerouting PINX for Global Call ID	6.8.26.2	T1:m	[ ]	Yes [ ] No[ ]
T5	Interactions for ANF-CIDL and ANF-PUMI at the Rerouting PINX for Thread ID	6.8.26.2	c.16	[ ]	Yes [ ] No[ ]

c.16: if (A5 and T1) then optional, else N/A

A.3.23 Interactions between ANF-CIDL and ANF-PUMO

Item	Question/feature	Reference	Status	N/A	Support
U1	Support of ANF-PUMO	ECMA-284	o		Yes [ ] No[ ]
U2	Interactions for ANF-CIDL and ANF-PUMO for Global Call ID	6.8.27	U1:m	[ ]	Yes [ ] No[ ]
U3	Interactions for ANF-CIDL and ANF-PUMO for Thread ID	6.8.27	c.17	[ ]	Yes [ ] No[ ]

c.17: if (A5 and U1) then optional, else N/A

A.3.24 Interactions between ANF-CIDL and SS-PUMR

Item	Question/feature	Reference	Status	N/A	Support
V1	Support of ANF-PUMR	ECMA-284	o		Yes [ ] No[ ]
V2	Interactions for ANF-CIDL and ANF-PUMR at the Home PINX for Thread ID	6.8.30.1	c.18	[ ]	Yes [ ] No[ ]
V3	Interactions for ANF-CIDL and ANF-PUMR at the Visitor PINX for Thread ID	6.8.30.2	c.18	[ ]	Yes [ ] No[ ]

c.18: if (A5 and V1) then optional, else N/A

A.3.25 Interactions between ANF-CIDL and SS-SSCT

Item	Question/feature	Reference	Status	N/A	Support
W1	Support of SS-SSCT	ECMA-300	o		Yes [ ] No[ ]
W2	Interactions for ANF-CIDL and SS-SSCT at the Rerouting PINX for Global Call ID	6.8.31	W1:m	[ ]	Yes [ ]
W3	Interactions for ANF-CIDL and SS-SSCT at the Rerouting PINX for Thread ID	6.8.31	c.19	[ ]	Yes [ ]

c.19: if (A5 and W1) then mandatory, else N/A

Annex B
(informative)

Examples of message sequences

This annex describes some typical message flows for ANF-CIDL. The following conventions are used in the figures of this annex:

1 The following notation is used:

➡	Message transporting ANF-CIDL specific operation
→	Message without ANF-CIDL specific operation
xxx.inv	Invoke APDU for operation xxx

2 The figures show messages exchanged via Protocol Control between PINXs involved in ANF-CIDL. Only messages relevant to ANF-CIDL are shown.

3 Only the relevant information content (i.e. remote operation APDUs) is listed below each message name. The Facility information elements containing remote operation APDUs are not explicitly shown. Information with no impact on ANF-CIDL is not shown.

4 The following abbreviations are used:

callId	callIdentification
CR	Call Reference
G	globalCallID
L	legID
T	threadID

B.1 Example message sequence for Generation of Call Identification Data – Generation at Call Establishment

Figure B.1 shows an example for the generation of Call Identification Data for a call at call establishment.

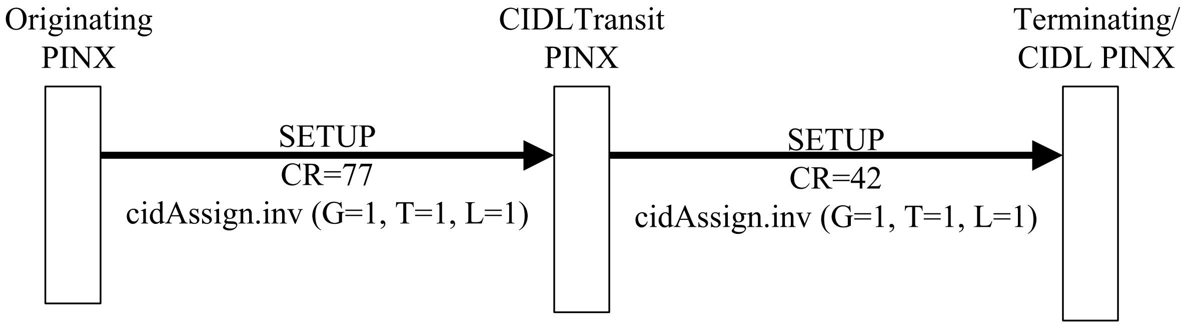

Figure B.1 - Message sequence for Generation of Call Identification Data – Generation at Call Establishment

B.2 Example message sequence for Update of Thread ID

Figure B.2 shows an example of the operation for the Update of the Thread ID where two calls are logically linked together.

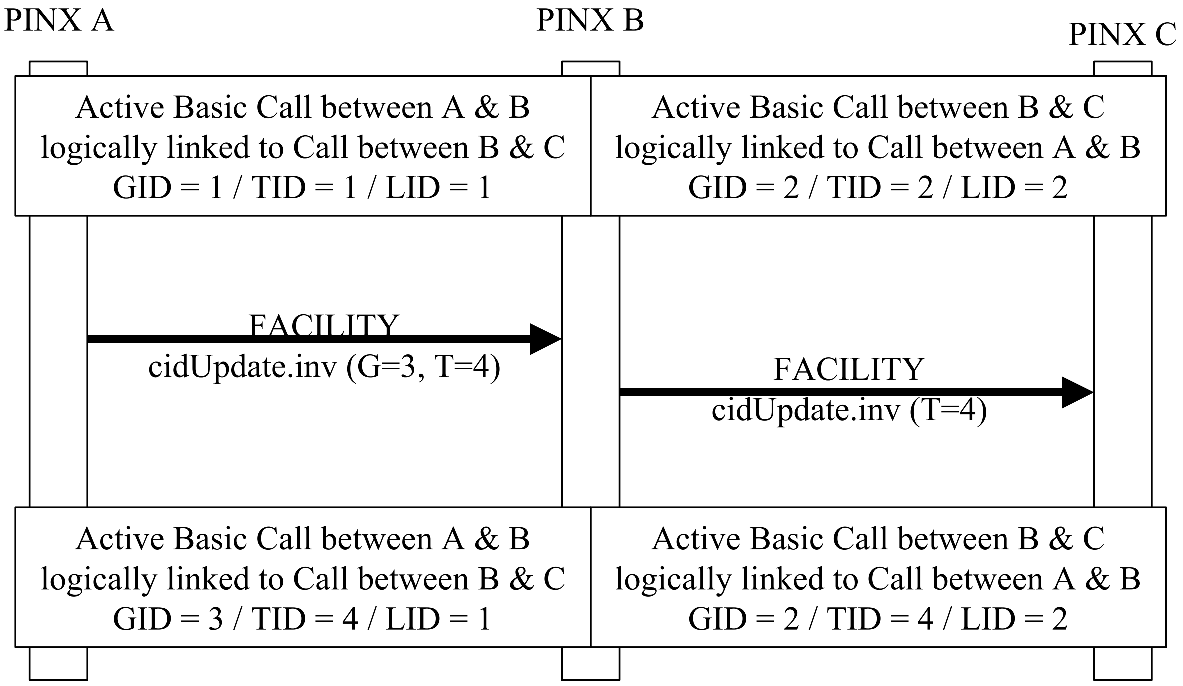
PINX A
PINX B
PINX C
Active Basic Call between A & B logically linked to Call between B & C GID = 1 / TID = 1 / LID = 1
Active Basic Call between B & C logically linked to Call between A & B GID = 2 / TID = 2 / LID = 2
FACILITY
cidUpdate.inv (G=3, T=4)
FACILITY
cidUpdate.inv (T=4)
Active Basic Call between A & B logically linked to Call between B & C GID = 3 / TID = 4 / LID = 1
Active Basic Call between B & C logically linked to Call between A & B GID = 2 / TID = 4 / LID = 2

Figure B.2 - Message sequence for Update of Thread ID

B.3 Example message sequence for Interaction with Call Transfer

Figure B.3 shows an example of interactions between the ANF-CIDL and the Call Transfer Supplementary Service.

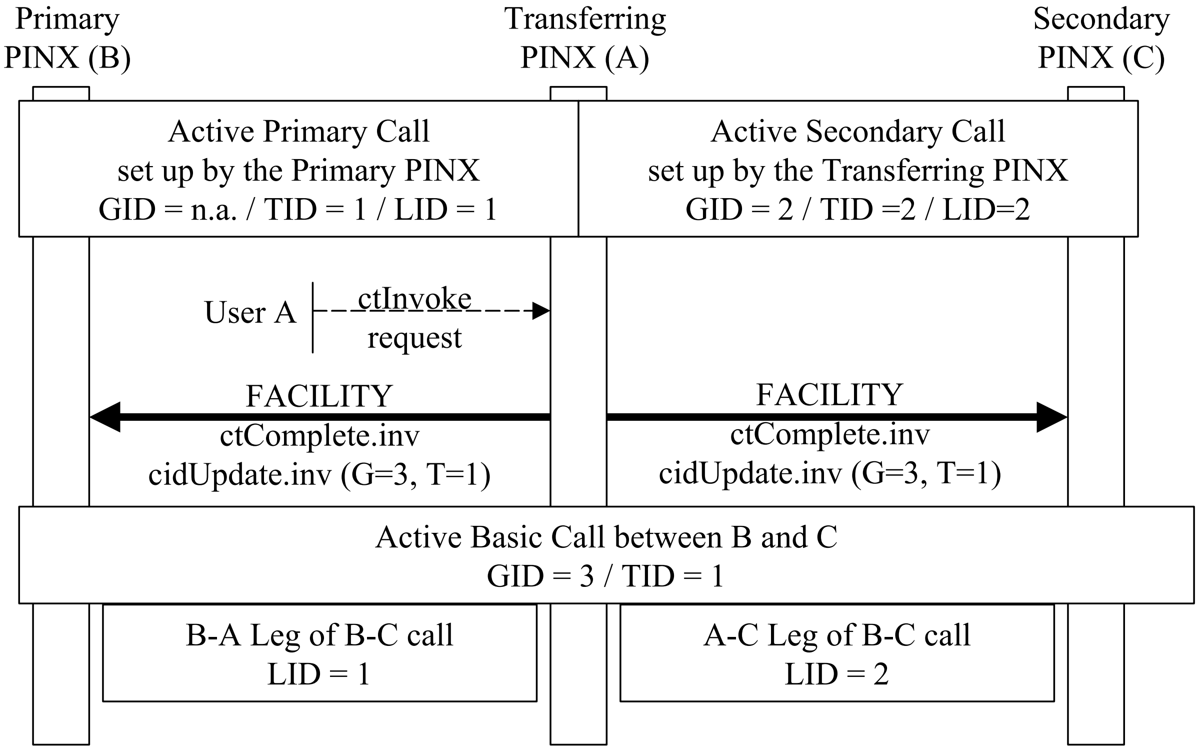
Primary PINX (B)
Transferring PINX (A)
Secondary PINX (C)
Active Primary Call set up by the Primary PINX GID = n.a. / TID = 1 / LID = 1
Active Secondary Call set up by the Transferring PINX GID = 2 / TID =2 / LID=2
User A
ctInvoke request
FACILITY ctComplete.inv cidUpdate.inv (G=3, T=1)
FACILITY ctComplete.inv cidUpdate.inv (G=3, T=1)
Active Basic Call between B and C GID = 3 / TID = 1
B-A Leg of B-C call LID = 1
A-C Leg of B-C call LID = 2

Figure B.3 - Message sequence for Interaction with Call Transfer

B.4 Example message sequence for Simultaneous Assignment

Figure B.4 shows an example of a ANF-CIDL Simultaneous Assignment, which is described in the following informative text for a better overview. For simplification, the values of GID, TID and LID are represented by number-values and notification flows (e.g. for invocation of Hold) are not shown within the figure.

⇒ User B sets up a call to User C, with GID, TID and LID all set to the value 1 (named “B-C call”).

⇒ User C invokes a consultation call (i.e. puts the B-C call on Hold) to User D (named “C-D call”), with GID and LID set to value 2 and the TID set to the value 1, as the consultation call is correlated to the B-C call.

⇒ User A sets up a call to User B, with GID, TID and LID all set to the value 2 (named “A-B call”). User B puts the B-C call on hold and accepts the call from User A.

⇒ User C invokes Call Transfer on the B-C and C-D calls. The two calls get updated with a new GID (value 5), but keep their TID (as already correlated due to the initial consultation C-D call) and also their initially assigned LIDs (as the LID is never changed by definition). PINX C enters CIDL-Simultaneous-Assignment-Supervision state.

⇒ At the same moment, user B invokes Call Transfer on the A-B and B-C calls. The two calls get updated with a new GID (value 4) and the B-C call additionally gets updated with the TID of the A-B call, as the A-B call is an incoming call. The two calls keep their initially assigned LIDs (as the LID is never changed by definition). PINX B enters CIDL-Simultaneous-Assignment-Supervision state.

⇒ Upon receipt of the callIdentificationUpdate invoke APDU with GID value 4 at PINX C from PINX B, PINX C performs (due to receipt of a callIdentificationUpdate invoke APDU in state CIDL-Simultaneous-Assignment-Supervision) the checks as described in 6.5.2.2, i.e.

- it determines that the stored GID (value 5) is higher then the received GID (value 4) and therefor updates the B-C call with GID value 5, i.e. it sends the stored (higher) GID value in a callIdentificationUpdate invoke APDU to PINX B;
- it determines that the received TID (value 3) is higher then the stored TID (value 1) and therefor stores the received (higher) TID value to the B-C and the C-D calls and updates the C-D call with TID value 3, i.e. it sends the new TID (value 3) in a callIdentificationUpdate invoke APDU to PINX D.

⇒ Upon receipt of the callIdentificationUpdate invoke APDU with GID (value 5) at PINX B from PINX C, PINX B performs (due to receipt of a callIdentificationUpdate invoke APDU in state CIDL-Simultaneous-Assignment-Supervision) the checks as described in 6.5.2.2, i.e.

- it determines that the stored GID (value 4) is lower then the received GID (value 5) and therefore stores the received (higher) GID value to the A-B and the B-C call and updates the A-B call with GID value 5, i.e. sends the new GID (value 5) in a callIdentificationUpdate invoke APDU to PINX A;
- it determines that the received TID (value 1) is lower then the stored TID (value 3) and therefor updates the B-C call with TID value 3, i.e. it sends the stored (higher) TID value in a callIdentificationUpdate invoke APDU to PINX C.

⇒ Upon receipt of the callIdentificationUpdate invoke APDU with GID (value 5) at PINX B from PINX C, PINX B performs (due to receipt of a callIdentificationUpdate invoke APDU in state CIDL-Simultaneous-Assignment) the checks as described in 6.5.2.2 and determines that the stored and received are identical and therefor does not perform any actions.

⇒ Upon receipt of the callIdentificationUpdate invoke APDU with TID (value 3) at PINX C from PINX B, PINX C performs (due to receipt of a callIdentificationUpdate invoke APDU in state CIDL-Simultaneous-Assignment) the checks as described in 6.5.2.2 and determines that the stored and received are identical and therefor does not perform any actions.

Due to the invocation of SS-CT at PINX B and PINX C, the scenario ends up in a call between User A and User D (“A-D call”), which has assigned end-to-end one GID (value 5) and one TID (value 3) but different

LIDs on the call legs of the originally initiated calls (i.e. LID value 3 for A-B leg, LID value 1 for B-C leg, LID value 2 for C-D leg).

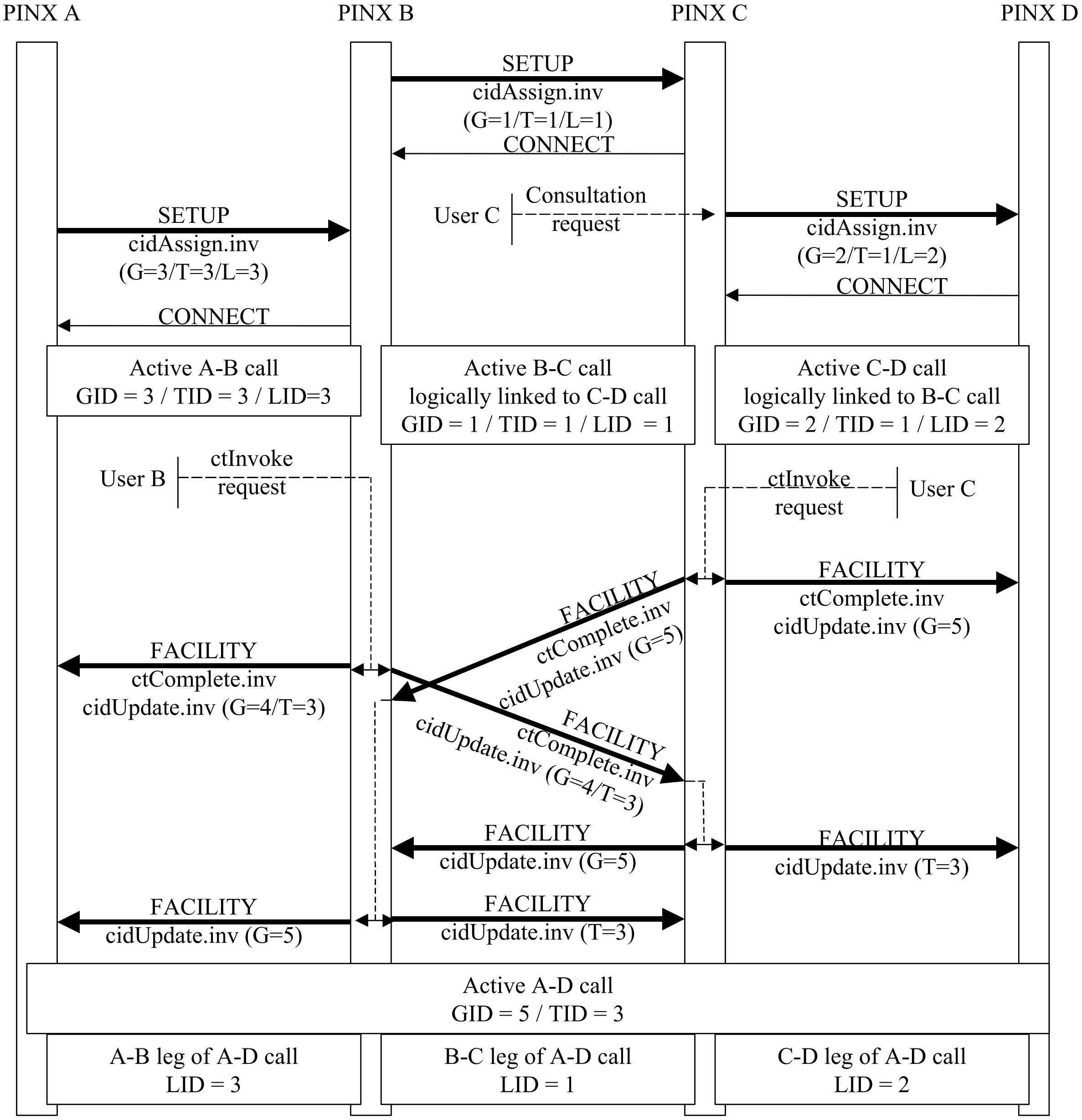

Figure B.4 - Message sequence for ANF-CIDL Simultaneous Assignment

Annex C
(informative)

Specification and Description Language (SDL) representation of procedures

The diagrams in this annex use the Specification and Description Language defined in ITU-T Rec. Z.100.

Each diagram represents the behaviour of a ANF-CIDL Supplementary Service Control entity at a particular type of PINX. In accordance with the protocol model described in ECMA-165, the Supplementary Service Control entity uses, via the Coordination Function, the services of Generic Functional Transport Control and Basic Call Control.

Where an output symbol represents a primitive to the Coordination Function, and that primitive results in a PSS1 message being sent, the output symbol bears the name of the message and any remote operation APDU(s) contained in that message. In case of a message specified in ECMA-143, basic call actions associated with the sending of that message are deemed to occur.

Where an input symbol represents a primitive from the Coordination Function, and that primitive results from a PSS1 message being received, the input symbol bears the name of the message and any remote operation APDU(s) contained in that message. In case of a message specified in ECMA-143, basic call actions associated with the receiving of that message are deemed to occur.

The following abbreviations are used:

inv.	invoke APDU
cidAssign	callIdentificationAssign
cidUpdate	callIdentificationUpdate

C.1 SDL Representation of ANF-CIDL at a Originating PINX

Figure C.1 shows the behavior of a ANF-CIDL Supplementary Service Control entity within a Originating PINX.

Output signals to the right represent primitives to the Coordination Function in respect of the messages being sent to the CIDL PINX.

Input signals from the left represent stimuli between the ANF-CIDL Supplementary Service Control entity and the ANF-CIDL user.

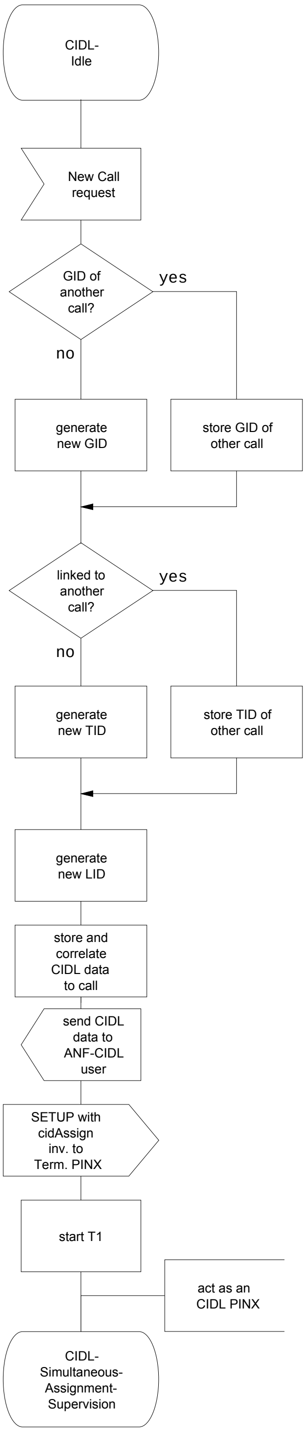

Figure C.1 – Originating PINX SDL

C.2 SDL Representation of ANF-CIDL at a CIDL PINX

Figure C.2 shows the behaviour of a ANF-CIDL Supplementary Service Control entity within a CIDL PINX.

Input signals from the left and output signals to the left represent primitives to and from the Coordination Function in respect of the messages being sent and received.

Input signals from the right and output signals to the right represent stimuli between the ANF-CIDL Supplementary Service Control entity and the ANF-CIDL user.

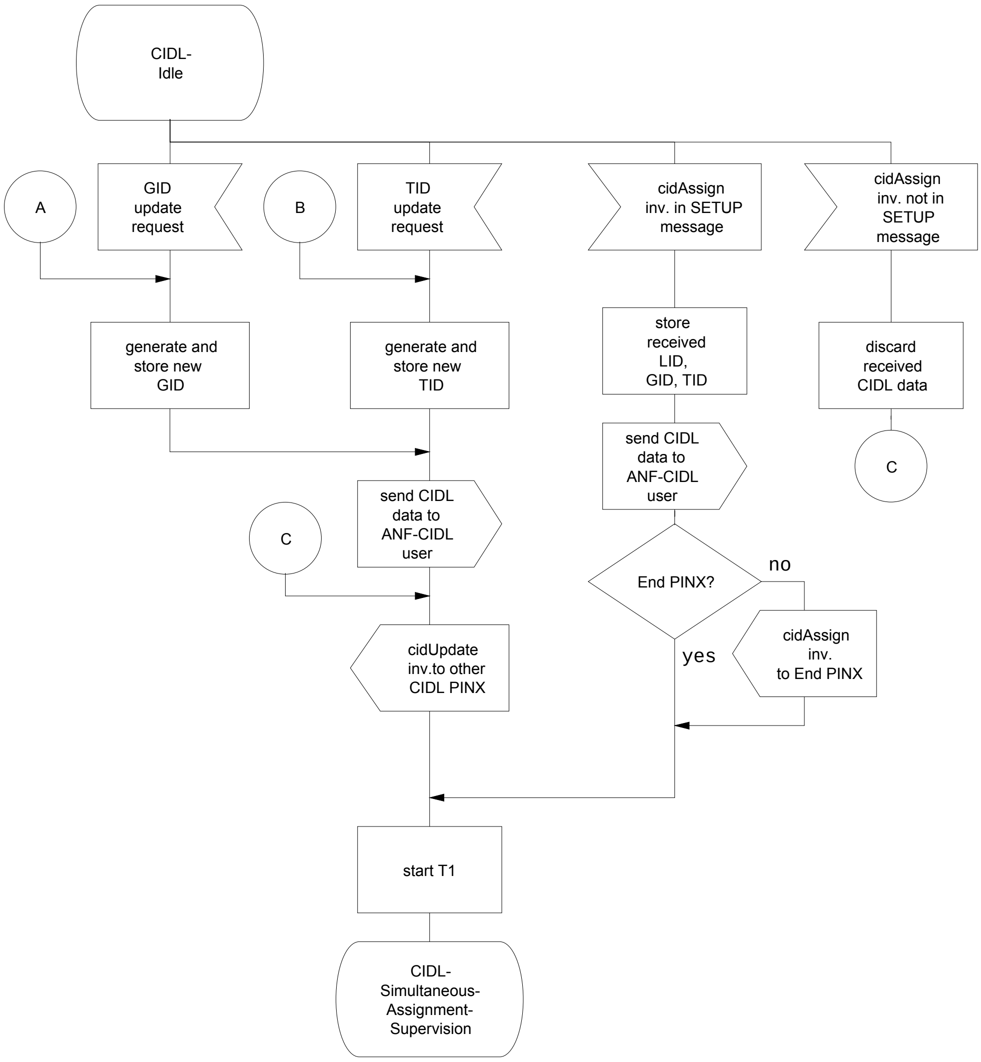
CIDL-
Idle
A
GID update request
B
TID update request
cidAssign inv. in SETUP message
cidAssign inv. not in SETUP message
generate and store new GID
generate and store new TID
store received LID, GID, TID
discard received CIDL data
send CIDL data to ANF-CIDL user
C
C
send CIDL data to ANF-CIDL user
End PINX?
no
yes
cidAssign inv. to End PINX
cidUpdate inv.to other CIDL PINX
start T1
CIDL-Simultaneous-Assignment-Supervision

Figure C.2 (sheet 1 of 3) - CIDL PINX SDL

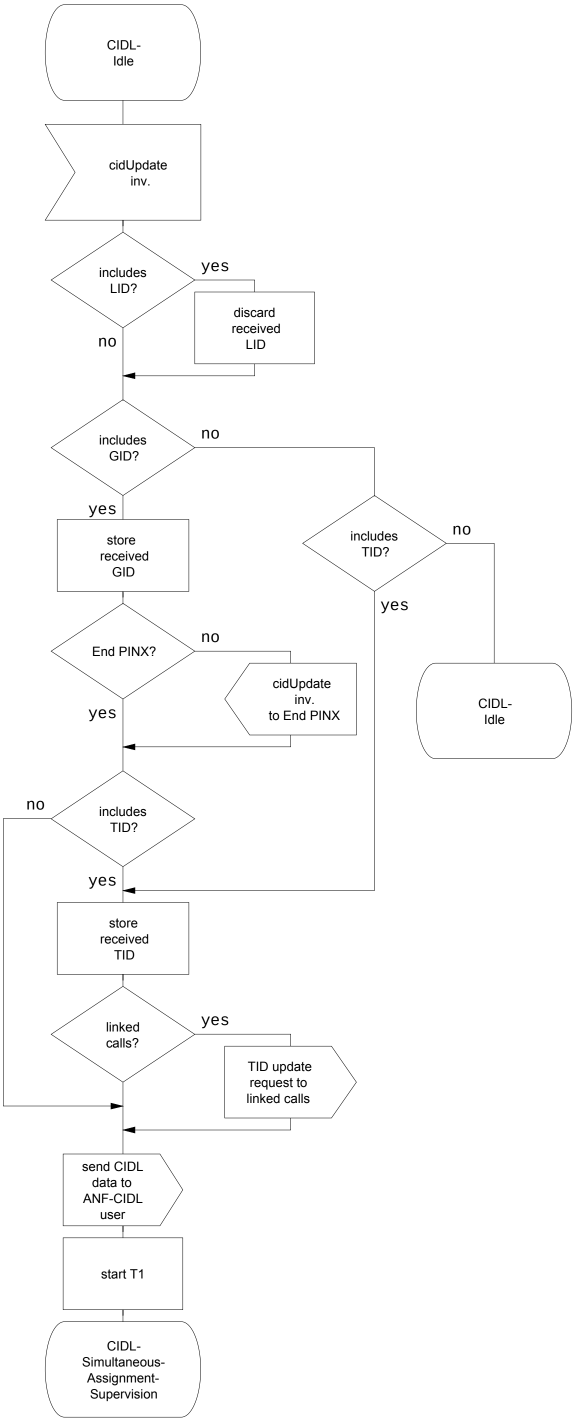

Figure C.2 (sheet 2 of 3) - CIDL PINX SDL

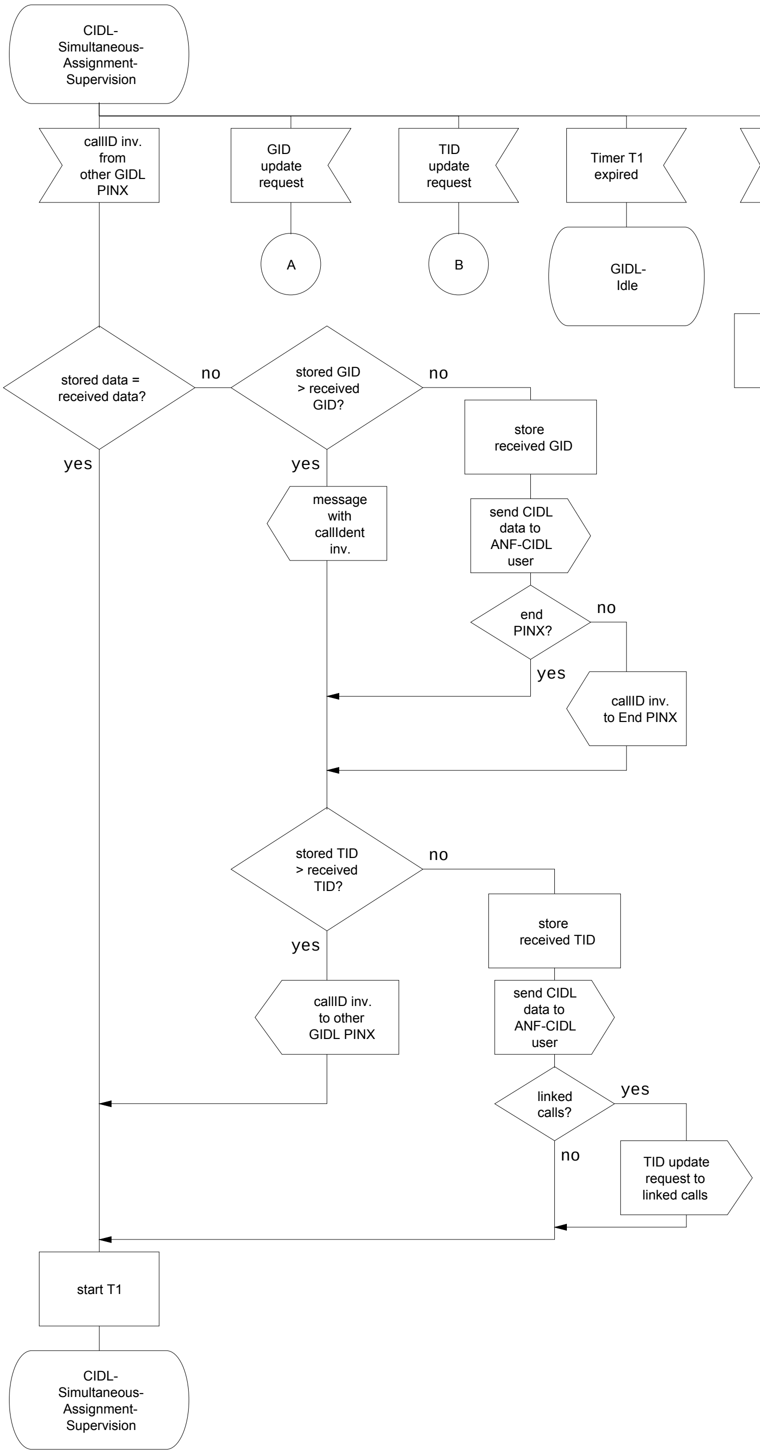
CIDL-Simultaneous-Assignment-Supervision
callID inv. from other GIDL PINX
GID update request
TID update request
Timer T1 expired
cidAssign inv. not in SETUP message
A
B
GIDL-Idle
discard received CIDL data
C
stored data = received data?
no
yes
stored GID > received GID?
no
yes
store received GID
send CIDL data to ANF-CIDL user
message with callIdent inv.
end PINX?
no
yes
callID inv. to End PINX
stored TID > received TID?
no
yes
store received TID
callID inv. to other GIDL PINX
send CIDL data to ANF-CIDL user
linked calls?
yes
no
TID update request to linked calls
start T1
CIDL-Simultaneous-Assignment-Supervision

Figure C.2 (sheet 3 of 3) – CIDL PINX SDL

C.3 SDL Representation of ANF-CIDL at a CIDL Transit PINX

Figure C.3 shows the behaviour of a ANF-CIDL Supplementary Service Control entity within a CIDL Transit PINX.

Input signals from the left and output signals to the left represent primitives to and from the Coordination Function in respect of the messages being sent and received.

Output signals to the right represent stimuli from the ANF-CIDL Supplementary Service Control entity to the ANF-CIDL user.

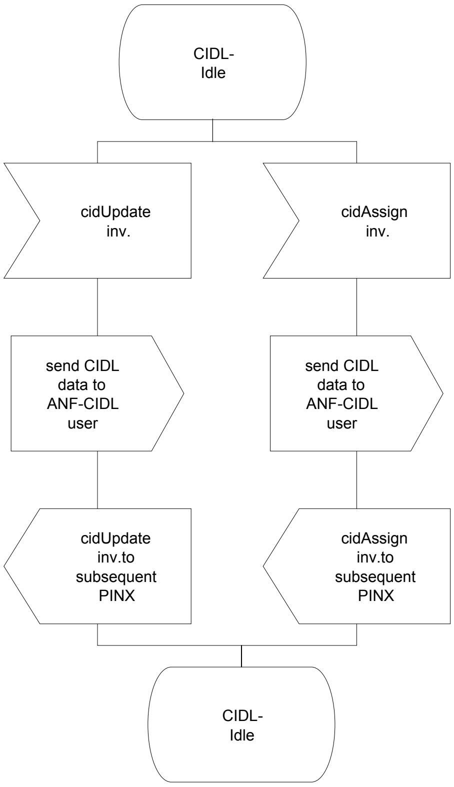

Figure C.3 – CIDL Transit PINX SDL

Annex D
(informative)

Imported definitions

This annex imports the text from ECMA-269 about how the elements of the CSTA Call Linkage Data has to be coded. As ECMA-269 refers to a specific part of ITU-T Recommendation H.225, also this part is imported.

References within the imported sub-clauses refer to the standards from which they were imported.

D.1 Import from ECMA-269, 12.2.5 CallLinkageData

The CallLinkageData parameter type contains globally unique information that is used to associate CSTA calls that are linked by a switching function because of the way that a call was created or manipulated.

For example, a call that is created as the result of the CSTA Consultation Call service (consulted call) may be linked to the call placed on hold at the consultation device by the CallLinkageData parameter type. The CallLinkageData parameter type can also be used to correlate calls in different switching sub-domains with the same "end-to-end" or "global" call.

See 6.1.4.7, "Call Linkage", on page 31 for a description of the call linkage feature.

Format

This parameter type is comprised of a sequence of the following:

- globalCallData (M) Structure - this component contains information that pertains to an "end-to-end" or "global" call. It consists of the following:
  - globalCallSwitchingSubDomainName (C) Characters - specifies the name of the switching sub-domain that created the globalCallData. The maximum size of this component is 64 characters. This component is required if the globalCallLinkageID component is not a globally unique value, otherwise it may be omitted. See Functional Requirement #2.
  - globalCallLinkageID (M) Choice - specifies the global call linkage identifier. This consists of one of the following choices:
    - subDomainCallLinkageID Octet String - specifies the switching function specific call linkage identifier. The maximum size of this string is 8 octets.
    - globallyUniqueCallLinkageID Octet String - specifies the globally unique call linkage identifier. The maximum size of this string is 16 octets. See functional requirement #2b.
  - callLinkageIDTimestamp (O) Timeinfo - specifies the time that the globalCallData was created. The format is described in 12.2.22.
- threadData (C) Structure - This information pertains to the entire call thread, i.e. all calls that are linked together as part of the same thread. This component shall only be provided if the switching function supports the thread linkage feature (as indicated by the capability exchange services). It consists of the following:
  - threadSwitchingSubDomainName (C) Characters - specifies the name of the switching sub-domain that created the threadData. The maximum size of this component is 64 characters. This is required if the threadID is not a globally unique value, otherwise it may be omitted. See Functional Requirement #2.
  - threadLinkageID (M) Choice - specifies the thread linkage identifier. This consists of one of the following choices:
    - subDomainThreadID Octet String - specifies the switching function specific thread linkage identifier. The maximum size of this string is 8 octets.

  - globallyUniqueThreadID Octet String - specifies the globally unique thread linkage identifier. The maximum size of this string is 16 octets. See functional requirement #2b.

- threadIDTimestamp (O) Timeinfo - specifies the time that the threadData was created. The format is described in 12.2.22.

Functional Requirements

1. The switching sub-domain names are used to distinguish one switching sub-domain from another. It is the responsibility of the switching function to provide a value that is unique within the switching domain.
2. The switching function that creates the callLinkageData ensures that it is globally unique by one of the following:
   a. providing the switching sub-domain name component along with a switching sub-domain unique identifier component. The combination of these two components provides call linkage data that is globally unique.

   b. providing a globally unique call linkage identifier component. The switching function that provides this choice of linkage identifier must generate the globallyUniqueCallLinkageID and/or the globallyUniqueThreadID via the algorithm used to create the globally unique ID as specified in H.225.
3. Whenever the globalCallData or the threadData is updated, the corresponding timestamp information shall also be updated.
4. When the switching function provides the globallyUniqueCallLinkageID choice, it shall ensure that:
   a. if the switching function also provides the globalCallSwitchingSubDomainName, there should be a 1/1 correspondence between the node name field in the globallyUniqueCallLinkageID and the globalCallSwitchingSubDomainName.

   b. if the switching function also provides the callLinkageIDTimestamp, the timestamp field in the globallyUniqueCallLinkageID should be consistent with the callLinkageIDTimestamp.
5. When the switching function provides the globallyUniqueThreadID choice, it shall ensure that:
   a. if the switching function also provides the threadSwitchingSubDomainName, there should be a 1/1 correspondence between the node name field in the globallyUniqueThreadID and the threadSwitchingSubDomainName.

   b. if the switching function also provides the threadIDTimestamp, the timestamp field in the globallyUniqueThreadID should be consistent with the threadIDTimestamp.

D.2 Import from ITU-T Recommendation H.225, section 7.6 H.225.0 common message elements

The GloballyUniqueID and ConferenceIdentifier are meant to be globally unique identifiers (GloballyUniqueID), the use of which is described in Recommendation H.323. A GloballyUniqueID is encoded with octet zero being encoded first. A GloballyUniqueID is formed according to table 16.

Table 16/H.225.0 – Globally unique ID formation

Field	Data type	Octet No.	Note
time_low	Unsigned 32-bit integer	0-3	The low field of the timestamp
time_mid	Unsigned 16-bit integer	4-5	The middle field of the timestamp
time_hi_and_version	Unsigned 16-bit integer	6-7	The high field of the timestamp multiplexed with the version number
clock_seq_hi_and_reserved	Unsigned 8-bit integer	8	The high field of the clock sequence multiplexed with the variant
clock_seq_low	Unsigned 8-bit integer	9	The low field of the clock sequence
node	Unsigned 48-bit integer	10-15	The spatially unique node identifier

The GloballyUniqueID consists of a record of 16 octets and must not contain padding between fields. The total size is 128 bits.

To minimize confusion about bit assignments within octets, the GloballyUniqueID record definition is defined only in terms of fields that are integral numbers of octets. The version number is multiplexed with the time stamp (*time_high*), and the variant field is multiplexed with the clock sequence (*clock_seq_high*).

The timestamp is a 60-bit value represented by Coordinated Universal Time (UTC) as a count of 100 nanosecond intervals since 00:00:00.00, 15 October 1582 (the date of Gregorian reform to the Christian calendar).

The version number is multiplexed in the 4 most significant bits of the *time_hi_and_version* field, and is set to 1 (binary 0001).

The variant field determines the layout of the GloballyUniqueID. The structure of a DCE GloballyUniqueID is fixed across different versions. Other GloballyUniqueID variants may not interoperate with a DCE GloballyUniqueID. Interoperability of GloballyUniqueIDs is defined as the applicability of operations such as string conversion, comparison, and lexical ordering across different systems. The *variant* field consists of a variable number of the MSBs of the *clock_seq_hi_and_reserved* field (see table 17).

Table 17/H.225.0 – Contents of the DCE variant field

msb1	msb2	msb3	Description
0	-	-	Reserved, NCS backward compatibility
1	0	-	DCE variant
1	1	0	Reserved, Microsoft Corporation GUID
1	1	1	Reserved for future definition

The clock sequence is required to detect potential losses of monotonicity of the clock. The clock sequence is encoded in the 6 least significant bits of the *clock_seq_hi_and_reserved* field and in the *clock_seq_low* field.

The *node* field consists of the IEEE address, usually the host address. For systems with multiple IEEE 802 nodes, any available node address can be used. The lowest addressed octet (octet number 10) contains the global/local bit and the unicast/multicast bit, and is the first octet of the address transmitted on an 802.3 packet-based network.

The clock sequence value should be changed whenever:

- The GloballyUniqueID generator detects that the local value of UTC has gone backward; this may be due to normal functioning of the DCE Time Service.

- The GloballyUniqueID generator has lost its state of the last value of UTC used, indicating that time may have gone backward; this is typically the case on reboot.

While a node is operational, the GloballyUniqueID generator always saves the last UTC used to create a GloballyUniqueID. Each time a new GloballyUniqueID is created, the current *UTC* is compared to the saved value and if either the current value is less (the non-monotonic clock case) or the saved value was lost, then the *clock sequence* is incremented modulo 16 384, thus avoiding production of duplicate GloballyUniqueIDs.

The *clock sequence* should be initialized to a random number to minimize the correlation across systems.

A GloballyUniqueID is generated according to the following algorithm:

1) Determine the values for the UTC-based timestamp and clock sequence to be used in the GloballyUniqueID.

2) Set the *time_low* field equal to the least significant 32-bits (bits numbered 0 to 31 inclusive) of the time stamp in the same order of significance.

3) Set the *time_mid* field equal to the bits numbered 32 to 47 inclusive of the time stamp in the same order of significance.

4) Set the 12 least significant bits (bits numbered 0 to 11 inclusive) of the *time_hi_and_version* field equal to the bits numbered 48 to 59 inclusive of the time stamp in the same order of significance.

5) Set the 4 most significant bits (bits numbered 12 to 15 inclusive) of the *time_hi_and_version* field to the 4-bit version number corresponding to the GloballyUniqueID version being created, as shown in table 17.

6) Set the *clock_seq_low* field to the 8 least significant bits (bits numbered 0 to 7 inclusive) of the *clock sequence* in the same order of significance.

7) Set the 6 least significant bits (bits numbered 0 to 5 inclusive) of the *clock_seq_hi_and_reserved* field to the 6 most significant bits (bits numbered 8 to 13 inclusive) of the *clock sequence* in the same order of significance.

8) Set the 2 most significant bits (bits numbered 6 and 7) of the clock_seq_hi_and_reserved to 0 and 1, respectively.

9) Set the node field to the 48-bit IEEE address in the same order of significance as the address.

If a system wants to generate a GloballyUniqueID but has no IEEE 802 compliant network card or other source of IEEE 802 addresses, then an alternative method should be used to generate a replacement value for the address. The ideal solution is to obtain a 47-bit cryptographic quality random number, and use it as the most significant 47 bits of the node ID, with the least significant bit of the first octet of the node ID set to 1. This bit is the unicast/multicast bit, which will never be set in IEEE 802 addresses obtained from network cards; hence, there can never be a conflict between GloballyUniqueIDs generated by machines with and without network cards.

If a system does not have a primitive to generate cryptographic quality random numbers, then in most systems there are usually a fairly large number of sources of randomness available from which one can be generated. Such sources are system specific, but often include the percent of memory in use, the size of main memory in bytes, the amount of free main memory in bytes, the size of the paging or swap file in bytes, free bytes of paging or swap file, the total size of user virtual address pace in bytes, the total available user address space bytes, the size of boot disk drive in bytes, the free disk space on boot drive in bytes, the current time, the amount of time since the system booted, the individual sizes of files in various system directories, etc.

For use in human readable text, a GloballyUniqueID string representation is specified as a sequence of fields, some of which are separated by single dashes.

Each field is treated as an integer and has its value printed as a zero-filled hexadecimal digit string with the most significant digit first. The hexadecimal values a to f inclusive are output as lower case characters, and are case insensitive on input. The sequence is the same as the GloballyUniqueID constructed type.

The formal definition of the GloballyUniqueID string representation is provided by the following extended BNF:

```
UUID                   = <time_low> <hyphen> <time_mid> <hyphen>
                         <time_high_and_version> <hyphen>
                         <clock_seq_and_reserved>
                         <clock_seq_low> <hyphen> <node>
time_low               = <hexOctet> <hexOctet> <hexOctet> <hexOctet>
time_mid               = <hexOctet> <hexOctet>
time_high_and_version  = <hexOctet> <hexOctet>
clock_seq_and_reserved = <hexOctet>
clock_seq_low          = <hexOctet>
node                   = <hexOctet><hexOctet><hexOctet>
                         <hexOctet><hexOctet><hexOctet>
hexOctet               = <hexDigit> <hexDigit>p
hexDigit               = <digit> | <a> | <b> | <c> | <d> | <e> | <f>
digit                  = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                         "8" | "9"
hyphen                 = "-"
a                      = "a" | "A"
b                      = "b" | "B"
c                      = "c" | "C"
d                      = "d" | "D"
e                      = "e" | "E"
f                      = "f" | "F"
```

The following is an example of the string representation of a GloballyUniqueID:

f81d4fae-7dec-11d0-a765-00a0c91e6bf6

Annex E
(normative)

ASN.1 definitions according to ITU-T Recs. X.208 / X.209

This annex lists all ASN.1 modules as they were defined in the first edition of ECMA-314, i.e. based on ITU-T Recommendations X.208 / X.209. Starting with the second edition the ASN.1 modules within ECMA-314 comply with ITU-T Recommendations X.680 / X.690. Please note that regardless of which version of these modules is used as a base of a QSIG implementation, the line encoding remains unchanged. Changes in future editions to modules based on X.680 / X.690 ASN.1 are not reflected in the modules in this annex.

Table E.1 - Call-Identification-and-Call-Linkage-Operations – based on ITU-T Recs. X.208 / X.209

```
Call-Identification-and-Call-Linkage-Operations
                        {iso(1) standard (0) pss1-call-identification-and-call-linkage (21889)
                        call-identification-and-call-linkage-operations (0)}

DEFINITIONS EXPLICIT TAGS ::=
BEGIN
IMPORTS
                   OPERATION
                   FROM  Remote-Operation-Notation
                   {joint-iso-ccitt(2) remote-operations(4) notation(0)}

                   Extension
                   FROM  Manufacturer-specific-service-extension-definition
                   {iso(1) standard(0) pss1-generic-procedures (11582)  msi-definition(0)};

CallIdentificationAssign       ::=     OPERATION
                                                 ARGUMENT        CallIdentificationAssignArg

CallIdentificationUpdate       ::=     OPERATION
                                                 ARGUMENT        CallIdentificationUpdateArg

CallIdentificationAssignArg    ::=   SEQUENCE {
                                       globalCallID      [0] CallIdentificationData,
                                       threadID          [1] CallIdentificationData  OPTIONAL,
                                       legID             [2] CallIdentificationData  OPTIONAL,
                                       extension         ExtensionType               OPTIONAL
                                       }

CallIdentificationUpdateArg    ::=   SEQUENCE {
                                       globalCallID      [0] CallIdentificationData  OPTIONAL,
                                       threadID          [1] CallIdentificationData  OPTIONAL,
                                       legID             [2] CallIdentificationData  OPTIONAL,
                                       extension         ExtensionType               OPTIONAL
                                       }
```

Table E.1 - Call-Identification-and-Call-Linkage-Operations – based on ITU-T Recs. X.208 / X.209 (concluded)

```
CallIdentificationData  ::= SEQUENCE {
-- this structure is according to ECMA-269, 12.2.5 (see annex D)
                              switchingSubDomainName  [0] IMPLICIT    SwitchingSubDomainName OPTIONAL,
                              linkageID               CHOICE {
                                  subDomainID             [1] IMPLICIT    SubDomainID,
                                  globallyUniqueID        [2] IMPLICIT    GloballyUniqueID},
                              timeStamp                   [3] IMPLICIT    TimeStamp OPTIONAL
                          }
SwitchingSubDomainName  ::=   IA5String (SIZE(1..64))

GloballyUniqueID  ::=     OCTET STRING (SIZE(1..16))
-- the GloballyUniqueID shall be coded according to ITU-T Recommendation H.225, section 7.6 (see annex D)
ExtensionType         ::=     CHOICE {
                              extension           [3] Extension,
                              sequenceOfExt       [4] IMPLICIT SEQUENCE OF Extension
                                        }

SubDomainID           ::=     OCTET STRING (SIZE(1..8))

TimeStamp             ::=     GeneralizedTime (SIZE(16..19))

callIdentificationAssign    CallIdentificationAssign   ::= localValue 105
callIdentificationUpdate    CallIdentificationUpdate   ::= localValue 106

END -- of Call-Identification-and-Call-Linkage-Operations
```

Free printed copies can be ordered from:
ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

Fax: +41 22 849.60.01
Email: documents@ecma.ch

Files of this Standard can be freely downloaded from the ECMA web site (www.ecma.ch). This site gives full information on ECMA, ECMA activities, ECMA Standards and Technical Reports.

ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

See inside cover page for obtaining further soft or hard copies.