

Release 8 CDR ASN.1 Format

PRINTOUT DESCRIPTION

Copyright

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

| | | |
|----------|---------------------|----------|
| 1 | Introduction | 1 |
| 2 | ASN.1 Format | 3 |





1 Introduction

This document presents the ASN.1 format for release 8 CDRs generated by the EPG.

For more information on CDRs, see [CDR Format for the GGSN and PGW](#) and [CDR Format for the SGW](#).





2 ASN.1 Format

GgsnPgwR8Ber DEFINITIONS IMPLICIT TAGS ::= BEGIN

GPRSRecord ::= CHOICE

```
{
  pgwRecord      [79] PGWRecord
}
```

PGWRecord ::= SET

```
{
  recordType                [0] RecordType,
  servedIMSI                [3] IMSI,
  p-GWAddress               [4] GSNAAddress,
  chargingID                [5] ChargingID,
  servingNodeAddress        [6] SEQUENCE OF GSNAAddress OPTIONAL,
  accessPointNameNI         [7] AccessPointNameNI OPTIONAL,
  pdpPDNType                [8] PDPTType OPTIONAL,
  servedPDPPDNAddress       [9] PDPAddress OPTIONAL,
  dynamicAddressFlag        [11] DynamicAddressFlag OPTIONAL,
  listOfTrafficVolumes      [12] SEQUENCE OF ChangeOfCharCondition OPTIONAL,
  recordOpeningTime         [13] TimeStamp,
  duration                  [14] CallDuration,
  causeForRecClosing        [15] CauseForRecClosing,
  recordSequenceNumber      [17] INTEGER OPTIONAL,
  nodeID                    [18] NodeID OPTIONAL,
  recordExtensions          [19] ManagementExtensions OPTIONAL,
  localSequenceNumber       [20] LocalSequenceNumber OPTIONAL,
  apnSelectionMode          [21] APNSelectionMode OPTIONAL,
  servedMSISDN              [22] MSISDN OPTIONAL,
  chargingCharacteristics    [23] ChargingCharacteristics,
  chChSelectionMode         [24] ChChSelectionMode OPTIONAL,
  imSsignalingContext       [25] NULL OPTIONAL,
  servingNodePLMNIdentifier [27] PLMN-Id OPTIONAL,
  pSFurnishChargingInformation [28] PSFurnishChargingInformation OPTIONAL,
  servedIMEISV              [29] IMEI OPTIONAL,
  rATType                   [30] RATType OPTIONAL,
  mSTimeZone                [31] MSTimeZone OPTIONAL,
  userLocationInformation    [32] OCTET STRING OPTIONAL,
  listOfServiceData         [34] SEQUENCE OF ChangeOfServiceCondition OPT
  servingNodeType           [35] SEQUENCE OF ServingNodeType OPTIONAL,
  p-GWPLMNIdentifier        [37] PLMN-Id OPTIONAL,
  startTime                  [38] TimeStamp OPTIONAL,
  stopTime                   [39] TimeStamp OPTIONAL,
  pdNConnectionID          [41] ChargingID OPTIONAL,
  threeGPP2UserLocationInformation [42] OCTET STRING OPTIONAL,
  servedPDPPDNAddressExt    [45] PDPAddress OPTIONAL
}
```



```
RecordType ::= INTEGER
{
  pGWRecord      (85)
}

AccessPointNameNI ::= IA5String (SIZE(1..63))
AddressString ::= OCTET STRING (SIZE (1..20))
AFChargingIdentifier ::= OCTET STRING
AFRecordInformation ::= SEQUENCE
{
  aFChargingIdentifier [1] AFChargingIdentifier
}
APNSelectionMode ::= ENUMERATED
{
  mSorNetworkProvidedSubscriptionVerified (0),
  mSProvidedSubscriptionNotVerified (1),
  networkProvidedSubscriptionNotVerified (2)
}
CallDuration ::= INTEGER
CauseForRecClosing ::= INTEGER
{
  normalRelease (0),
  abnormalRelease (4),
  volumeLimit (16),
  timeLimit (17),
  servingNodeChange (18),
  maxChangeCond (19),
  rATChange (22),
  mSTimeZoneChange (23),
  sGSNPLMNIDChange (24),
  managementInitRelease (100),
  creditControlChange (102),
  creditControlInitRelease (104),
  policyControlInitRelease (105),
  zoneInfoChange (110)
}
ChangeCondition ::= ENUMERATED
{
  qosChange (0),
  tariffTime (1),
  recordClosure (2),
  failureHandlingContinueOngoing (3),
  failureHandlingRetryandTerminateOngoing (4),
  failureHandlingTerminateOngoing (5)
}
ChangeOfCharCondition ::= SEQUENCE
{
  qosNegotiated [2] QoSInformation OPTIONAL,
  dataVolumeGPRSuplink [3] DataVolumeGPRS,
  dataVolumeGPRSDownlink [4] DataVolumeGPRS,
```



```

    changeCondition          [5] ChangeCondition,
    changeTime               [6] TimeStamp,
    userLocationInformation  [8] OCTET STRING OPTIONAL,
    ePCQoSInformation        [9] EPCQoSInformation OPTIONAL
}
ChangeOfServiceCondition ::= SEQUENCE
{
    ratingGroup              [1] RatingGroupId,
    resultCode               [3] ResultCode OPTIONAL,
    localSequenceNumber      [4] LocalSequenceNumber OPTIONAL,
    timeOfFirstUsage         [5] TimeStamp OPTIONAL,
    timeOfLastUsage         [6] TimeStamp OPTIONAL,
    timeUsage                [7] CallDuration OPTIONAL,
    serviceConditionChange   [8] ServiceConditionChange,
    qoSInformationNeg        [9] EPCQoSInformation OPTIONAL,
    servingNodeAddress       [10] GSNAddress OPTIONAL,
    datavolumeFBCUplink      [12] DataVolumeGPRS OPTIONAL,
    datavolumeFBCDownlink    [13] DataVolumeGPRS OPTIONAL,
    timeOfReport             [14] TimeStamp,
    failureHandlingContinue  [16] FailureHandlingContinue OPTIONAL,
    serviceIdentifier        [17] ServiceIdentifier OPTIONAL,
    pSFurnishChargingInformation [18] PSFurnishChargingInformation OPTIONAL,
    aFRecordInformation      [19] SEQUENCE OF AFRecordInformation OPTIONAL,
    userLocationInformation  [20] OCTET STRING OPTIONAL,
    eventBasedChargingInformation [21] EventBasedChargingInformation OPTIONAL,
    threeGPP2UserLocationInformation [24] OCTET STRING OPTIONAL
}
ChargingCharacteristics ::= OCTET STRING (SIZE (2))
ChargingID ::= INTEGER (0..4294967295)
ChChSelectionMode ::= ENUMERATED
{
    servingNodeSupplied (0),
    homeDefault (3),
    radiusSupplied (100),
    roamingClassBased (101)
}
DataVolumeGPRS ::= INTEGER
DynamicAddressFlag ::= BOOLEAN
EPCQoSInformation ::= SEQUENCE
{
    qCI                      [1] INTEGER,
    maxRequestedBandwithUL   [2] INTEGER OPTIONAL,
    maxRequestedBandwithDL   [3] INTEGER OPTIONAL,
    guaranteedBitrateUL      [4] INTEGER OPTIONAL,
    guaranteedBitrateDL      [5] INTEGER OPTIONAL,
    aRP                      [6] INTEGER OPTIONAL
}

EventBasedChargingInformation ::= SEQUENCE
{
    numberOfEvents [1] INTEGER,

```



```
    eventTimeStamps [2] SEQUENCE OF TimeStamp OPTIONAL
  }
  FailureHandlingContinue ::= BOOLEAN
  FFDApendIndicator ::= BOOLEAN
  FreeFormatData ::= OCTET STRING (SIZE (1..160))
  GSNAaddress ::= IPAddress
  IMEI ::= TBCD-STRING (SIZE (8))
  IMSI ::= TBCD-STRING (SIZE (3..8))
  IPAddress ::= IPBinaryAddress
  IPBinaryAddress ::= CHOICE
  {
    ipBinV4Address [0] OCTET STRING (SIZE(4)),
    ipBinV6Address [1] OCTET STRING (SIZE(16))
  }
  ISDN-AddressString ::= AddressString (SIZE(1..9))
  LocalSequenceNumber ::= INTEGER (0..4294967295)
  ManagementExtensions ::= SET OF ManagementExtension
  ManagementExtension ::= SEQUENCE
  {
    identifier OBJECT IDENTIFIER,
    significance [1] BOOLEAN DEFAULT TRUE,
    information [2] GprsCdrExtensions
  }
  MSISDN ::= ISDN-AddressString
  MSTimeZone ::= OCTET STRING (SIZE (2))
  NodeID ::= IA5String (SIZE(1..20))
  PDPAddress ::= CHOICE
  {
    IPAddress [0] IPAddress
  }
  PDPTYPE ::= OCTET STRING (SIZE(1..2))
  PLMN-Id ::= OCTET STRING (SIZE(3))
  PSFurnishChargingInformation ::= SEQUENCE
  {
    pSFreeFormatData [1] FreeFormatData,
    pSFFDApendIndicator [2] FFDApendIndicator OPTIONAL
  }
  QoSInformation ::= OCTET STRING (SIZE (4..15))
  RatingGroupId ::= INTEGER
  RATType ::= INTEGER (0..255)
  ResultCode ::= INTEGER
  ServiceConditionChange ::= BIT STRING
  {
    qosChange (0),
    sGSNChange (1),
    sGSNPLMNIDChange (2),
    tariffTimeSwitch (3),
    pDPContextRelease (4),
    rATChange (5),
    configurationChange (8),
    serviceStop (9),
  }
```



```

dCCATimeThresholdReached      (10),
dCCAVolumeThresholdReached    (11),
dCCAServiceSpecificUnitThresholdReached (12),
dCCATimeExhausted             (13),
dCCAVolumeExhausted           (14),
dCCAValidityTimeout           (15),
dCCAReauthorisationRequest    (17),
dCCAContinueOngoingSession    (18),
dCCARetryAndTerminateOngoingSession (19),
dCCATerminateOngoingSession   (20),
cGI-SAICChange                (21),
rAICChange                     (22),
dCCAServiceSpecificUnitExhausted (23),
recordClosure                  (24),
eCGIChange                    (29),
tAICChange                    (30)
}
ServiceIdentifier ::= INTEGER (0..4294967295)
TBCD-STRING ::= OCTET STRING
TimeStamp ::= OCTET STRING (SIZE(9))

GprsCdrExtensions ::= SET
{
  creditControlInfo [2] CreditControlInfo OPTIONAL,
  policyControlInfo [3] PolicyControlInfo OPTIONAL,
  userCategory [5] INTEGER OPTIONAL,
  ruleSpaceId [6] IA5String OPTIONAL,
  serviceContainers [7] SEQUENCE OF ServiceContainer OPTIONAL,
  timeReports [8] SEQUENCE OF TimeReport OPTIONAL,
  dataZoneInfo [9] IA5String OPTIONAL,
  creditStatus [10] INTEGER OPTIONAL
}

ActiveTimeMethod ::= ENUMERATED
{
  duration (1),
  inactivityIncluded (2),
  inactivity (3),
  activePeriods (4)
}

CreditControlFailureReport ::= SEQUENCE
{
  requestType [0] CreditRequestType,
  requestStatus [1] CreditRequestStatus,
  resultCode [2] CreditResultCode OPTIONAL,
  ccRequestNumber [12] INTEGER OPTIONAL
}

CreditControlInfo ::= SEQUENCE
{
  creditControlFailureReport [6] CreditControlFailureReport OPTIONAL,
  creditControlSessionId [7] OCTET STRING (SIZE(1..255)) OPTIONAL,

```



```
        ccsRealm                [8] OCTET STRING (SIZE(1..255)) OPTIONAL
    }
    CreditRequestType ::= ENUMERATED
    {
        start    (0),
        interim  (1),
        stop     (2)
    }
    CreditRequestStatus ::= ENUMERATED
    {
        unsent      (0),
        noAnswer    (1),
        failure     (2),
        oCSRestart  (3)
    }
    CreditResultCode ::= INTEGER
    PolicyControlFailureReport ::= SEQUENCE
    {
        requestType    [0] PolicyRequestType,
        requestStatus  [1] PolicyRequestStatus,
        resultCode     [2] PolicyResultCode OPTIONAL,
        stopTime       [5] TimeStamp OPTIONAL
    }
    PolicyControlInfo ::= SEQUENCE
    {
        policyControlFailureReport [4] PolicyControlFailureReport OPTIONAL,
        pcsRealm                   [6] OCTET STRING (SIZE(1..255)) OPTIONAL,
        policyControlSessionId     [7] OCTET STRING (SIZE(1..255)) OPTIONAL
    }
    PolicyRequestType ::= ENUMERATED
    {
        start    (0),
        interim  (1),
        stop     (2)
    }
    PolicyRequestStatus ::= ENUMERATED
    {
        unsent      (0),
        noAnswer    (1),
        failure     (2),
        pCRFRestart (3)
    }
    PolicyResultCode ::= INTEGER
    RatingGroup ::= INTEGER
    ServiceContainer ::= SEQUENCE
    {
        ratingGroup          [1] RatingGroup,
        serviceIdentifier    [2] ServiceIdentifier OPTIONAL,
        localSequenceNumber [3] LocalSequenceNumber OPTIONAL,
        method               [4] ActiveTimeMethod OPTIONAL,
        inactivity           [5] INTEGER OPTIONAL,
    }
```



```

    resolution          [6] INTEGER OPTIONAL,
    ccRequestNumber     [7] INTEGER OPTIONAL,
    serviceSpecificUnits [8] INTEGER OPTIONAL,
    listOfURI           [9] SEQUENCE OF URI OPTIONAL,
    listOfIPFlow        [10] SEQUENCE OF IPFlow OPTIONAL
  }
  ServingNodeType ::= ENUMERATED
  {
    sGSN                (0),
    pMIPSGW             (1),
    gTPSGW              (2),
    ePDG                (3),
    hSGW                (4),
    mME                 (5),
    tWAN                (6)
  }
  TimeReport ::= SEQUENCE
  {
    ratingGroup         [1] RatingGroup,
    startTime           [2] TimeStamp,
    endTime             [3] TimeStamp,
    dataVolumeUplink    [4] DataVolumeGPRS OPTIONAL,
    dataVolumeDownlink [5] DataVolumeGPRS OPTIONAL
  }
  URI ::= SEQUENCE
  {
    count               [1] INTEGER OPTIONAL,
    uri                 [2] IA5String OPTIONAL,
    uriIdentifier        [3] INTEGER OPTIONAL,
    uriDataVolumeUplink [4] INTEGER OPTIONAL,
    uriDataVolumeDownlink [5] INTEGER OPTIONAL,
    listOfUriTimeStamps [6] SEQUENCE OF TimeStamp OPTIONAL
  }
  IPFlow ::= SEQUENCE
  {
    uEAddress           [1] IPAddress OPTIONAL,
    uEPort              [2] IPPort OPTIONAL,
    networkAddress      [3] IPAddress OPTIONAL,
    networkPort         [4] IPPort OPTIONAL,
    protocol             [5] IPProtocol OPTIONAL,
    dataVolumeUplink    [6] INTEGER OPTIONAL,
    dataVolumeDownlink [7] INTEGER OPTIONAL,
    listOfURI           [8] SEQUENCE OF UTF8String OPTIONAL
  }
  IPPort ::= INTEGER (0..65535)
  IPProtocol ::= INTEGER (0..255)
  END

```