

# ACR Storage in MTAS

INTERWORK DESCR

© Ericsson AB 2008-2013

## Disclaimer Statements

No part of this document may be reproduced in any form without the written permission of the copyright owner.

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 damages of any kind resulting from the use of this document.

## Trademark List

All trademarks are the property of their respective owners.

## Contents

<b>1</b>	<b>Document History.....</b>	<b>4</b>
<b>2</b>	<b>Scope and Purpose .....</b>	<b>4</b>
2.1	Interface Entities .....	4
2.2	Interface Role.....	6
2.3	Services .....	6
2.4	Encapsulation and Addressing.....	6
<b>3</b>	<b>Procedures.....</b>	<b>7</b>
3.1	Overview .....	7
3.2	Lower Level Procedures.....	7
3.3	Fetching of ACR Files .....	8
<b>4</b>	<b>Information Model.....</b>	<b>8</b>
4.1	General .....	8
<b>5</b>	<b>Formal Syntax or Schema .....</b>	<b>8</b>
5.1	ACR File Format .....	8
5.1.1	ACR File Header Format .....	9
5.1.2	ACR Record Header Format .....	14
5.1.3	Diameter Header Format.....	15
5.1.4	ACRs.....	15
5.1.5	ACR File Naming Convention .....	15
<b>6</b>	<b>Related Standards.....</b>	<b>16</b>
<b>7</b>	<b>Terminology .....</b>	<b>16</b>
7.1	Abbreviations .....	16

7.2        Definitions ..... 17

8         **References..... 17**

# 1 Document History

Rev	Date	Sign	Comment
A	2013-03-06	eleerik	Clone from 5/155 19-CRA 119 2107 rev A New document numbers used from MTAS 13A FD1
C	2013-09-20	erudhes	Added Local Storage of MCID information
D	2014-04-23	ekorrssa	MTAS 14B <ul style="list-style-type: none"> <li>- The version of the IP address (IPv4 or IPv6) in the file header now depends on the mtasSiplpVersion CM parameter.</li> <li>- HS51026: Clarification added about Number of ACR field in the File Header.</li> </ul>
E	2014-12-01	eptebar	Generalized version that is platform agnostic.
F	2015-11-10	edinjia	MTAS 4.0.0 <ul style="list-style-type: none"> <li>- Removed the description about the Number of ACRs is always 0 when the files closed in an abnormal way.</li> </ul>
G	2016-03-23	xaliraa	Updated References section : CBA link to MOM
H	2016-04-20	edandwe	MTAS 4.2.0 Editorial correction: Deleted erroneous listings of preliminary revisions.
J	2017-05-22	ezhayic	MTAS 4.7/MTASv 1.7 HV74371. Description of file transfer function has been updated to meet platform agnostic.

## 2 Scope and Purpose

### 2.1 Interface Entities

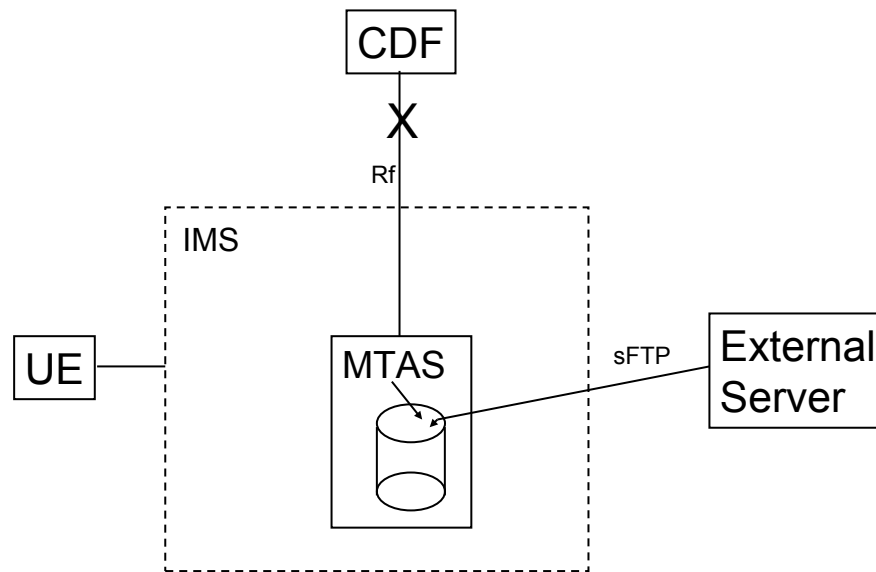


Figure 1: Interface Entities

The Offline Charging interface, Rf interface, is defined by 3GPP and is the reference point between a Charging Trigger Function (CTF) and a Charging Data Function (CDF).

MTAS acts as a CTF, collecting the information pertaining to changeable events, assembling this information into matching charging events and sending these charging events in Offline Charging Accounting Request (ACR) messages towards the CDF.

In case of a failure situation when a primary CDF and a secondary (if applicable) can not be reached, the MTAS will if the ACR Storage function is enabled, write the Offline Charging ACR messages into ACR files on file server.

The ComDetails interface is a diameter based interface towards the Communication Details Server. If the local storage function is enabled, the MCID information is stored locally in ACR files on file server, instead of being reported to the Communication Details Server via the ComDetails interface.

The stored ACR files follow a specific ACR file format described in 5.1, and they can be fetched later and transferred to an external server by the use of the file transfer procedure described in [3] .

The purpose of this document is to specify how the files can be fetched and to describe the format and encoding of the ACR files. It also describes the handling and naming of the files.

## 2.2 Interface Role

This document describes the services offered by MTAS that are used by an external server.

In failure situations when MTAS cannot reach a CDF the ACR Storage function may (depending on configuration data) be triggered to store ACR information to disk.

This document describes how the ACR files are fetched and the format and encoding of the stored ACR files in order for an external server to be able to decode the content of the ACR files after retrieving the files from MTAS by the use of sFTP.

## 2.3 Services

*Table 1: Offered Services*

Offered Service	Description
Storing of Offline Charging ACR data in ACR files.	Offline Charging ACR information is stored in ACR files according to a specific format described in chapter 5.1. The information is only stored if MTAS cannot reach any CDF or if an ACA is received with an error result code. The ACR files can be retrieved by an external server by the use of sFTP.
Storing of MCID ACR data in ACR files	The MCID ACR information is stored in ACR files according to a specific format described in chapter 5.1. The information is stored if the Local Storage function is enabled. The MCID ACR files can be retrieved by an external server by the use of sFTP.

*Table 2: Used Services*

Used Service	Description
None	

## 2.4 Encapsulation and Addressing

The Offline Charging ACR files are stored on the file system. ACR files that are closed and ready to be transferred are given a filename as described in chapter 5.1.5.

The MCID ACR files are stored on the local file system. MCID ACR files that are closed and ready to be transferred are given a filename as described in chapter 5.1.5.

The stored ACR Files can be retrieved via sFTP by the use of the file transfer procedure. For more information see [3].

## 3 Procedures

### 3.1 Overview

If the ACR storage function is enabled, MTAS will request the ACR Storage Handler to store ACR data when MTAS is unable to send an ACR message to a CDF due to failure of the connection between MTAS and the CDF. Detailed information on when an ACR is sent for ACR storage is described in [2].

### 3.2 Lower Level Procedures

The ACR files are stored on the file server.

When an ACR file is closed the file is renamed from a temporary file name to a permanent file name as specified in chapter 5.1.5. When a permanent file name has been designated to the file it may be retrieved by the external server.

A file is closed when any of the following conditions are fulfilled:

- A configurable file size limit as specified by CM attribute `mtasChargingAcrStorageMaxFileSize` has been reached.
- At expiry of a timer that is started when the first ACR is stored in a file. The timer is set to the value specified by the CM attribute `mtasChargingAcrStorageMaxTime`. This timer will be restarted each time a new file has been opened and the first ACR is to be stored.
- When the number of ACRs in the file reaches the max limit specified by the CM attribute `mtasChargingAcrStorageMaxNbrAcr`.
- The disk used for ACR storage becomes full.
- At a request to terminate the process that is handling the ACR storage, all currently opened files are closed.

- After a restart has occurred if there are any open files they are closed and the “File Closure Trigger Reason” field in the ACR file header is set to 128 Abnormal file closure (Undefined error closure reason) to indicate that the content can not be fully trusted.

When ACRs are started to be stored an “ACR disk storage started” alarm is raised.

The alarm is cleared when there are no ACR files stored any longer in any file server.

### 3.3 Fetching of ACR Files

The file transfer procedure described in [3] can be used for retrieving the stored ACR files, which transfers the ACR files from the disks to an external server.

The management of the file transfer is described in [3].

## 4 Information Model

### 4.1 General

Depending on the underlying platform, MTAS offers possibility for ACR files to be transferred to an external server.

The management of the file transfer is described in [3].

## 5 Formal Syntax or Schema

### 5.1 ACR File Format

ACR files consist of a variable length ACR File Header followed by a variable number of (zero or more) ACR records. Each individual ACR consists of an ACR Record Header followed by a Diameter Header, followed by Attribute Value Pairs (AVPs).



ACR File Header
ACR Record Header
Diameter Header
AVP1
AVP2
ACR Record Header
Diameter Header
AVP1
AVP2
.....

*Figure 2: ACR File content*

The ACR is written in binary encoded format.

## 5.1.1 ACR File Header Format

The ACR File Header contains information about the ACR records in the ACR file. As the content of the ACR File Header are based on the ACR records in the ACR file, the ACR File Header is populated after the last ACR record is included and the ACR file is ready to be closed.

Figure 3 below illustrates the contents of the ACR File Header.

	Bits							
Octets	8	7	6	5	4	3	2	1
1..4	File Length							
5..8	Header Length							
9	High Release Identifier				High Version Identifier			
10	Low Release Identifier				Low Version Identifier			
11..14	File Opening Timestamp							
15..18	Time stamp when last ACR was appended the file							
19..22	Number of ACRs in the file							
23..26	File sequence number							
27	File Closure Trigger Reason							
28..47	IP address of Node that generated the file							
48	Lost ACR indicator							
49..50	Length of ACR routing filter							

*Figure 3: ACR file format*

By default, all fields are mandatory and are always included in the ACR File Header. Exceptions are included in the individual field descriptions below.

### 5.1.1.1 File Length

The “file length” field made up of 4 octets contains a binary value that indicates the total length of the ACR file in octets, including the file header and the total ACR payload length.

### **5.1.1.2 Header Length**

The “Header Length” field, made up of 4 octets, contains a binary value that indicates the total length of the ACR file header in octets.

### **5.1.1.3 High Release Identifier / High Version identifier**

The High Release Identifier field contains the highest release of ACR records in the ACR file. The high Version Identifier field contains the highest version of the highest release of ACR records in the ACR file. The highest release identifier is stored on the most significant 3 bits of the octet, and the highest version number is stored on the least significant 5 bits of the same octet.

MTAS supports V9.5.0 of TS 32.260 reference [4]. The value included in these fields is always 110 that is related to release 9 for High Release Identifier and 00101 that relates to version 5 for High Version Identifier in ACR files created by the MTAS.

### **5.1.1.4 Low Release Identifier / Low Version Identifier**

The Low Release Identifier field contains the lowest release of ACR records in the ACR file. The Low version Identifier field contains the lowest version of the lowest release of ACR records in the ACR file. The lowest release number is stored on the most significant 3 bits of the octet, and the lowest version number is stored on the least significant 5 bits of the same octet.

For more information, see [4]. The value indicated in these fields is always 100 for low Release Identifier and 00110 for Low Version Identifier in ACR files created by the MTAS.

### **5.1.1.5 File opening timestamp**

The File Opening Timestamp field, made up of 4 octets, contains the time the ACR file was opened. The time values are spread out over 4 octets according to the format described below:

- Binary bits 1 to 4 indicate the month (1..12) according to the CTF's local time zone;
- Binary bits 5 to 9 indicate the day (1..31) according to the CTFs local time zone;
- Binary bits 10 to 14 contain the hour (0..23), according to the CTF local time zone;
- Binary bits 15 to 20 indicate the minute (0..59) according to the CTFs local time zone;

- Binary bit 21 indicates the sign of the deviation of the CTF's local time from Universal Time Coordinated (UTC). A bit value of 1 expresses + time deviation, while a bit value of 0 expresses – time deviation. In case there is no time deviation from UTC, the binary bit can take either value.
- Binary bits 22 to 26 indicate the hour (0..23) deviation of the CTF's local time from UTC
- Binary bits 27 to 32 indicate the minute (0..59) deviation of the CTF's local time from UTC.

#### 5.1.1.6 Last ACR append timestamp

The Last ACR Append Timestamp field made up of 4 octets, contains the time the last ACR was appended to the ACR file. The time values are spread out over 4 octets according to the format described below:

- Binary bits 1 to 4 indicate the month (1..12) according to the CTF's local time zone;
- Binary bits 5 to 9 indicate the day (1..31) according to the CTFs local time zone;
- Binary bits 10 to 14 contain the hour (0..23), according to the CTF local time zone;
- Binary bits 15 to 20 indicate the minute (0..59) according to the CTFs local time zone;
- Binary bit 21 indicates the sign of the deviation of the CTF's local time from Universal Time Coordinated (UTC). A bit value of 1 expresses + time deviation, while a bit value of 0 expresses – time deviation. In case there is no time deviation from UTC, the binary bit can take either value.
- Binary bits 22 to 26 indicate the hour (0..23) deviation of the CTF's local time from UTC
- Binary bits 27 to 32 indicate the minute (0..59) deviation of the CTF's local time from UTC

#### 5.1.1.7 Number of ACRs in file

This parameter made up of 4 octets contains a binary value that specifies the total number of ACR records that are included in the ACR file.

#### **5.1.1.8 File sequence number**

The File Sequence Number field, made up of 4 octets, contains a binary value that identifies the running number of the ACR files generated by the CTF.

The first ACR file generated by the CTF carries the sequence number 1, and each additional ACR file generated increases the running number by 1. The sequence number is reset to “1” at the start of each outage, or when the running number reaches the maximum value (all binary bits are set to 1).

#### **5.1.1.9 File Closure Trigger Reason**

The File Closure Reason field provides a means to determine the reason that the file was closed by the CTF. It is encoded as a single octet as follows:

Normal closure reasons (binary value 0 to 127)

0=Normal closure (undefined normal closure reason).

1=File size limit reached (O&M configured)

2=File open-time limit reached (O&M configured)

3=Maximum number of ACRs in file reached (O&M configured)

4=File closed by manual intervention

5=ACR release, version or encoding change

6 to 127 are reserved for future use

It is expected that when problems occur with the File System, that it may not be possible to write an abnormal closure reason, but if possible here are the abnormal closure reasons (binary values 128 to 255):

128=Abnormal file closure

129=File System Error

130=File system storage exhausted

131=File integrity error

132 to 255 are reserved for future use.

Values 129-155 are not used by MTAS.

#### 5.1.1.10 IP address of node that generated the file

This parameter indicates the OAM VIP address of the MTAS generating the file. For both IPv4 and IPv6 addresses, the parameter is encoded in IPv6 representation. The first four bytes of the parameter, which are preceding this IPv6 address, are insignificant and filled with 'FF'.

The used IP address reflects the setting for the `mtasSipIpAddress` CM parameter. For more information, regarding this CM parameter, see [6].

MTAS encodes the parameter in IPv6 address representation.

#### 5.1.1.11 Lost ACR indicator

This parameter, made up of 1 octet, contains a binary value which indicates if and how many ACRs were lost during the processing in the CTF. The term "lost" implies that the ACR(s) could not be placed into the destination file due to irrecoverable errors.

Appropriate indication is given according to the following encoding of the "lost" ACR indicator".

- MSB bit "0", all other bits "0": no ACRs have been lost;
- MSB bit "0"; all other bits set to a value corresponding to decimal 1 to decimal 126: MTAS has identified that a number of ACRs corresponding to the value of the lower 7 bits were lost, while it is unknown whether more ACR were lost. This value is not used by MTAS.
- MSB bit "0", all other bits set to "1": MTAS has identified that 127 or more ACRs were lost, while it is unknown while it is unknown whether more ACRs were lost; This value is not used by MTAS.
- MSB bit "1", all other bits "0": ACRs have been lost but MTAS cannot determine the number of lost ACRs; This value is used by MTAS if a file is still open after a restart has occurred as the content of the file can not be fully trusted.
- MSB bit "1" all other bits set to a value corresponding to decimal 1 to decimal 126: MTAS has calculated the number of lost ACRs as indicated in the value of the lower 7 bits;
- MSB bit "1", all other bits set to "12: MTAS has calculated the number of lost ACRs to be 127 or more.

#### 5.1.1.12 Length of ACR Routing filter

This parameter contains a binary value that specifies the length of the subsequent ACR routing filter in octets.

This field is not supported in ACR files created by the MTAS. The binary value is set to 0.

## 5.1.2 ACR Record Header Format

Each individual ACR record consists of an ACR Record Header, followed by a Diameter Header, followed by AVPs.

The format of an ACR Record header is given in the table below:

Octets	Bits							
	8	7	6	5	4	3	2	1
1..2	ACR length							
3	Release identifier				Version Identifier			
4	Data Record Format				TS number			

*Figure 4: ACR Record Header Format*

### 5.1.2.1 ACR Length

This two octet field contains a binary value that specifies the length of the subsequent ACR, excluding the 4 header octets.

### 5.1.2.2 Release Identifier

This three bit field contains a binary value that identifies the 3GPP Release of the TS 32.260 that is supported by MTAS. MTAS uses the value 6 that is 110 in binary format that corresponds to release 9.

See [5] for a detailed description of how the Release Identifier is defined.

### 5.1.2.3 Version identifier

This five bit field contains a binary value that identifies the version of the TS 32.260. MTAS supports version “5” that is value 00101 in binary format.

### 5.1.2.4 Data Record Format

This three bit field contains a binary value that identifies the ACR encoding It is set to “5” that signifies the use of ACR Data File.

### 5.1.2.5 TS number

This five bit field contains a binary value that identifies the number of the TS ACR encoding and is set to “9” that is 01001 in binary format that identifies TS 32.260.

### 5.1.3 Diameter Header Format

The Diameter Header Format is as described in [1].

### 5.1.4 ACRs

The ACRs stored in the ACR files are the same as the ones used in generic ACR Offline Charging.

For a complete description of the AVPs used in the Offline Charging ACRs refer to [2].

The ACRs are stored in Diameter format.

### 5.1.5 ACR File Naming Convention

Before a file is closed it is given a file name that includes following information:

<NodeID>\_<RC>.<date>\_<time>

1. NodeID is the configured Node Distinguished Name of the MTAS that generated the file.
2. The RC parameter is a running count, starting with a value of "1". Note that the delimiter preceding this field is made up of an underscore character (\_) followed by a minus character (-), followed by another underscore character (\_). The RC will have the same value as the File Sequence Number field in the ACR File Header described in 5.1.1.8.
3. The "date" field indicates in ASCII, the date when the ACR file was closed. It is of the form YYYYMMDD, where:
  - YYYY is the year in four-digit notation
  - MM is the month in two digit notation (01-12)
  - DD is the day in two-digit notation (01-31)

Note that this field is preceded by a point (.) character as delimiter.

4. The "time" field indicates in ASCII, the time when the ACR file was closed. It is of the form HHMMshhmm, where:
  - HH is the two-digit hour of the day (local time), based on 24-hour clock (00-23);
  - MM is the two digit minute of the hour (local time) in two digit notation (00...59);

- s is in ASCII, the sign of the deviation of the CTF local time from Universal Time Coordinated (UTC) (+ or -). In case there is no time deviation from UTC, is -.
- hh is the hour deviation between local time and UTC in two digit notation (00-23);
- mm is the minute deviation between local time and UTC in two digit notation (00-59).

Note that the delimiter preceding this field is made up of an underscore character (\_) followed by a minus character (-), followed by another underscore character (-).

The following example shows the file name of ACR file number 44 the CTF named MTASEAST01 located in time zone UTC +1 would generate on the 5<sup>th</sup> of May 2012 at 18:00.

**MTASEast01\_-\_44.20120505\_-\_1800+0100**

An ACR file that has been given a name as described above is ready to be retrieved.

## 6 Related Standards

3GPP Technical Specification; Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging  
3GPP TS 32.260 (v9.5.0).

3GPP TS 32.297 v7.0.0 CDR File Format and Transfer.

## 7 Terminology

### 7.1 Abbreviations

ACR	Accounting Request
AVP	Attribute-Value Pair
CDF	Charging Data Function
CTF	Charging Trigger Function
CM	Configuration Management
MCID	Malicious Communication Identification
MMTel	Multimedia Telephony



## 7.2 Definitions

# 8 References

RFC3588, Diameter Base Protocol, <http://www.ietf.org/rfc/rfc3588.txt>

IWD Diameter Offline Charging in MTAS 13/15519-AVA 901 18

TSP: File Transfer Utility User Guide 1/1553-CRA 119 0002/2

CBA: File Management 6/1551-APA 901 44/1

3GPP Technical Specification; Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging  
3GPP TS 32.260 (v9.5.0).

3GPP TS 32.297 v7.0.0 CDR File Format and Transfer

TSP: 1/190 84-AVA 901 09/n \*\* MTAS, Parameter Description

CBA: 155 54-LZN 765 0163/n \*\* Managed Object Model MTAS

(\*\* See the Customer or Support library for the Application System in question)