Configuration Note

AudioCodes Routing Manager (ARM)

Generating Call Detail Records

using ARM CDR Generator

Version 10.0

AudioCodes Routing Manager

audiocodes

Notice

Date Published: April-02-2025

Security Vulnerabilities

All security vulnerabilities should be reported to vulnerability@audiocodes.com.

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

Stay in the Loop with AudioCodes

Related Documentation

Manual Name
ARM Release Notes
ARM Installation Manual

Manual Name
ARM User's Manual
ARM REST API Developer's Guide
Mediant 9000 SBC User's Manual
Mediant 4000 SBC User's Manual
Mediant 3000 Gateway User's Manual
Mediant 2600 E-SBC User's Manual
Mediant SE SBC User's Manual
Mediant SE-H SBC User's Manual
Mediant VE SBC User's Manual
Mediant VE-H SBC User's Manual
Mediant 1000B Gateway and E-SBC User's Manual
Mediant 800B Gateway and E-SBC User's Manual
Mediant 500 Gateway and E-SBC User's Manual
Mediant 500 MSBR User's Manual
Mediant 500L Gateway and E-SBC User's Manual
Mediant 500L MSBR User's Manual
MP-1288 High-Density Analog Media Gateway User's Manual
One Voice Operations Center Server Installation, Operation and Maintenance Manual
One Voice Operations Center Integration with Northbound Interfaces
One Voice Operations Center User’s Manual
One Voice Operations Center Product Description
One Voice Operations Center Alarms Guide
One Voice Operations Center Security Guidelines

Document Revision Record

LTRT	Description
42100	Initial document release.
42101	One Start-Call; CdrMessage > CdrArmMessage; sessionKey; routingRuleId; routingRuleName; discardingByRoutingRule; partial; description; /opt/tomcat/logs/CDR_Json; drop-calls period; myOutputFileName; UnfinishedCdrs.json
42102	New CdrArmMessage fields. -iu option. -od option.
42103	Updated to 10.0.200 (Score).

Table of Contents

1 Introduction

This document explains how to implement the AudioCodes Routing Manager (ARM) CDR Generator.

The CDR Generator is a utility that converts CdrMessages into CdrArmMessages. The conversions are performed in the following CdrArmMessages formats:

- Clear text
- JSONs

CdrMessages are sent by the Router for each leg which takes part in a call.

CdrMessages are of two types:

- Start call
- End call

A CdrArmMessage contains the total quantity of information about a call taken from the CdrMessage. In general, a successful call (without an alternative route) has three CdrMessages:

- One Start-Call
- Two End-Call

The CDR Generator takes all these messages and gathers them into one message.

To display the CDR in more readable format, you can generate a CSV file and define the column name of each CDR field.

2 Configuring CdrArmMessage Fields

The following below describes the CDR ARM message fields. Use the table as a reference when configuring the Map file for the output CSV file.

Table 2-1: CdrArmMessage Field Descriptions

CDR Field	Description
CdrApplicationType	Defines the node application type: ■ SBC ■ GW ■ HYBRID ■ THIRD_PARTY
sessionKey	Defines a unique key identifier.
sessionId	Defines a unique session identifier.
nodeId	Defines the ARM node's database ID.
nodeName	Defines the node name as described in the ARM GUI.
nodeIp	Defines the node's IP address.
incomingPconOrConnectionName	Defines the incoming leg name as described in the ARM GUI.
incomingSipInterface	Defines the SIP interface ID of an incoming Connection or Peer Connection in the SBC / Gateway.
incomingCallId	Defines the Call ID of the incoming leg.
outgoingPconOrConnectionName	Defines the outgoing leg name as described in the ARM GUI.
outgoingCallId	Defines the Call ID of the outgoing leg.
srcUri	Defines the Source URI that is sent (following manipulation).
srcUriBeforeMap	Defines the Source URI before manipulation.
from	Defines the From URI that is sent (following

CDR Field	Description
	manipulation).
fromBeforeMap	Defines the From URI before manipulation.
pai	Defines the P-Asserted-Identity URI that it sent (following manipulation).
paiBeforeMap	Defines the P-Asserted-Identity URI before manipulation.
ppi	Defines the P-Prefetred-Identity URI sent (following manipulation).
ppiBeforeMap	Defines the P-Prefetred-Identity URI before manipulation.
dstUri	Defines the Destination URI sent (following manipulation).
dstUriBeforeMap	Defines the Destination URI before manipulation.
armSetupTime	Defines the time at which CALL_START is sent, per the ARM's time.
armReleaseTime	Defines the time at which CALL_END is sent, per the ARM's time.
sbcSetupTime	Defines the the time at which Gateway / SBC time when starting to handle an Invite message, as reported by the Gateway / SBC.
sbcConnectTime	Defines the time at which a 200 OK response is sent (i.e., when the call is established), as reported by the Gateway / SBC.
sbcReleaseTime	Defines the time at which a BYE message is sent (i.e., when a call ends), as reported by the Gateway / SBC.
sbcAlertTime	Defines the time at which the remote side phone starts to ring, as reported by the Gateway / SBC.
alertDuration	Defines how long the phone rings, in milliseconds, as reported by the Gateway / SBC. The SBC / Gateway must be configured to send the time (duration) in milliseconds.

CDR Field	Description
voiceDuration	Defines how long voice is streamed, in milliseconds, as reported by the Gateway / SBC. The SBC / Gateway must be configured to send the time (duration) in milliseconds.
completeDuration	Defines how long the entire call takes, from the first incoming Invite until the call is ended, in milliseconds, as reported by the Gateway / SBC.
voiceStreamed	Determines whether the call was established or not.
sipTerminationReason	Defines the SIP termination reason.
sipTerminationReasonDesc	Defines a more detailed and descriptive SIP termination reason than the field 'sipTerminationReason'.
routeSeq	Defines the number that each route (path) of a call has, starting from 0.
lastNodeId	Defines the ARM database ID of the last node in the path.
lastNodeName	Defines the name of the last node in the path as described in the ARM GUI.
lastPconOrConnectionName	Defines the name of the last Peer Connection or Connection in the path.
routingRuleId	Defines the Routing Rule ID of the matching rule
routingRuleName	Defines the Routing Rule name of the matching rule
discardingByRoutingRule	Defines the Routing Rule ID in case of discarding rule
path	Describes the path.
partial	Defines whether all CdrMessages applicable to this route are found in the input file. Will be ‘True’ for in-progress calls.
description	Briefly describes a partial call.

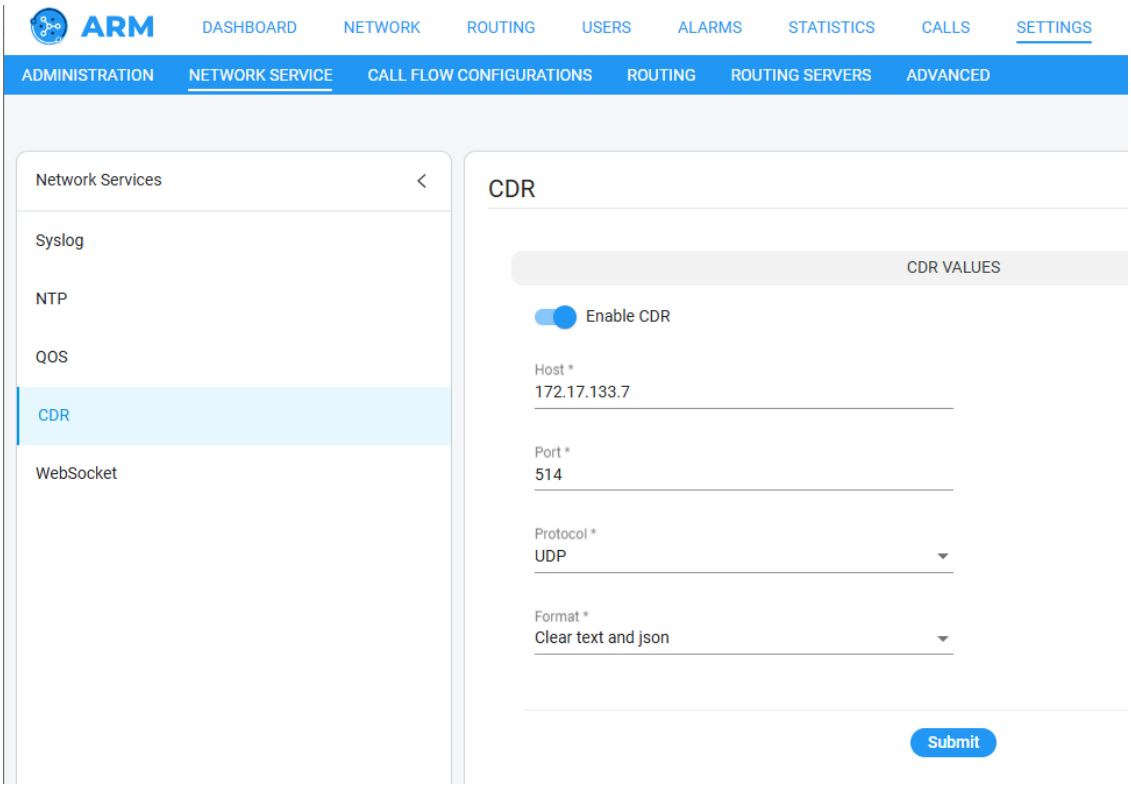
CDR Field	Description
score	The score value that is returned from SecureLogix.

3 Generating CDRs

ARM allows you to generate CDRs using the CDR Generator utility.

➢ **To generate CDRs using the CDR Generator utility:**

1. Open the CDR page in the ARM GUI (**Settings** > **Network Service** > **CDR**).
2. Click the **Enable CDR** toggle button to enable CDR generation, and then from the 'Format' drop-down list, select **Json** or **Clear text and Json**:



3. Collect all the CDRs, using one of the following methods:
   - If you have a syslog server, take the syslog file from the syslog server.
   - If you do not have a syslog server, collect the logs from All Routers and create one large file: ARM saves the CDRs in JSON format under the /opt/tomcat/logs/CDR_Json directory. The name of the CDR file is CDR_Json-*.log (up to 10 files can be saved). Create one file: 'CDR_Json.log'.
4. Generate a CdrArmMessages file from CDRs:
   a. Copy the one CDR file you got into one of the Routers.
   b. Run the following command for generating the CDR file:

```
cdrUtil -i CDR_Json.txt
```

   The output files are:

- CdrArm_ CDR_Json _[timestamp].txt
- CdrArm_ CDR_Json _[timestamp]. json

5. To set the output file name, use the -o option. For example:

```
cdrUtil -i CDR_Json.log -o myOutputFileName
```

6. To set the drop-calls period in hours, use the -d option. For example:

```
cdrUtil  -d 12 CDR_Json.log
```

All partial calls created more than 12 hours before are exported to the file 'DroppedCdrs.json'. The other partial calls are part of the input CDR in the next cdrUtil that is run and are exported to the file 'UnfinishedCdrs.json'.

When running cdrUtil again, it uses the 'myOutputFileName' input and the file 'UnfinishedCdrs.json'.

'UnfinishedCdrs.json' is overwritten with new cdrUtil results.

The file 'UnfinishedCdrs.json' will then no longer be used as input and each cdrUtill UnfinishedCdrs message will be added to it.

The default value for drop-calls is 24 hours.

7. To set the input 'UnfinishedCdrs.json' file location, use the -iu option. For example:

```
cdrUtil CDR_Json.log -iu /root/UnfinishedCdrs.json
```

8. To set the output directory, use the -od option. For example:

```
cdrUtil CDR_Json.log -od /root/tempDir
```

The output directory will contain the following files:

- CdrArm_ CDR_Json _[timestamp].txt
- CdrArm_ CDR_Json _[timestamp]. Json
- UnfinishedCdrs.json
- DroppedCdrs.json'

9. To set the date format, use the -f option. For example:

```
cdrUtil CDR_Json.log -f "yyyy-MM-dd HH:mm:ss.SSSSZZ"
```

10. Generate a CSV file of the CdrArmMessages file from CDRs. To generate a CSV file, use the -m option. For example:

```
cdrUtil -i CDR_Json.log -m myMapFileName
```

The map file is a text file in json format that defines which fields will be displayed in the CSV file and what their new name (if required by the customer) will be. The file contains a list of name pairs:

- armCdrFieldName [the name of the field from the CdrArmMessage]
- convertedCdrFieldName [the customer-required name of the field that will be displayed in the CSV file]

- The generated CSV file contains only the 'armCdrFieldName' field from the mapping json file but you can dynamically build a CSV file, customize CSV file column names and determine where the CSV file will take its columns from. Use the example in the next section as reference.
- To create a new column with a new name which isn't displayed in the CdrArmMessage, set 'armCdrFieldName' to empty and 'convertedCdrFieldName' to the new column name. See convertedCdrFieldName = Direction in the section below.

4 Map File Example

Use the example below as reference if you need to dynamically build a CSV file. Note the list of name pairs contained in the file:

- "armCdrFieldName" [the name of the field from the CdrArmMessage]
- "convertedCdrFieldName" [the customer-required name of the field that will be displayed in the CSV file]

```
{
  "name": "Puzzel",

  "cdrFieldNameJsonList": [
    {
      "armCdrFieldName": "srcUriBeforeMap",
       "convertedCdrFieldName": "ANUM before Manipulation"
    },
    {
      "armCdrFieldName": "srcUri",
       "convertedCdrFieldName": "ANUM after Manipulation"
    },
    {
      "armCdrFieldName": "dstUriBeforeMap",
       "convertedCdrFieldName": "BNUM before Manipulation"
    },
    {
      "armCdrFieldName": "dstUri",
       "convertedCdrFieldName": "BNUM after Manipulation"
    },
    {
      "armCdrFieldName": "sbcConnectTime",
       "convertedCdrFieldName": "START"
    },
    {
      "armCdrFieldName": "sbcReleaseTime",
       "convertedCdrFieldName": "FINISH"
    },
    {
      "armCdrFieldName": "nodeName",
       "convertedCdrFieldName": "EXI"
    },
    {
        "armCdrFieldName": "lastNodeName",
       "convertedCdrFieldName": "FXI"
```

```
    },
    {
      "armCdrFieldName": "incomingPconOrConnectionName",
        "convertedCdrFieldName": "ITI"
    },
    {
      "armCdrFieldName": "lastPconOrConnectionName",
        "convertedCdrFieldName": "OTI"
    },
    {
      "armCdrFieldName": "sipTerminationReason",
        "convertedCdrFieldName": "Diagnostic"
    },
    {
      "armCdrFieldName": "voiceStreamed",
        "convertedCdrFieldName": "ToBeCharged"
    },
    {
      "armCdrFieldName": "",
        "convertedCdrFieldName": "Direction"
    },
    {
      "armCdrFieldName": "incomingCallId",
        "convertedCdrFieldName": "SipCallId"
    },
    {
      "armCdrFieldName": "sbcSetupTime",
        "convertedCdrFieldName": "SetupAt"
    },
    {
      "armCdrFieldName": "sbcAlertTime",
        "convertedCdrFieldName": "AlertAt"
    },
    {
      "armCdrFieldName": "completeDuration",
        "convertedCdrFieldName": "msSetup"
    },
    {
      "armCdrFieldName": "alertDuration",
        "convertedCdrFieldName": "msAlert"
    }
    ]
}
```

This page is intentionally left blank.

International Headquarters

6 Ofra Haza Street

Naimi Park

Or Yehuda, 6032303, Israel

Tel: +972-3-976-4000

Fax: +972-3-976-4040

AudioCodes Inc.

80 Kingsbridge Rd

Piscataway, NJ 08854, USA

Tel: +1-732-469-0880

Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide

Website: https://www.audiocodes.com/

Documentation Feedback: https://online.audiocodes.com/documentation-feedback

Document #: LTRT-42103