Multi-Service Business Gateways

VoIP Media Gateways

SIP, MGCP, MEGACO, TPNCP Protocols

Configuration Note
Debug Recording Feature

July 2011

Document # LTRT-28701

AudioCodes

Table of Contents

List of Figures

List of Tables

Notice

This document describes the Debug Recording functionality.

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee the accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document and other documents can be viewed by registered customers at http://www.audiocodes.com/downloads.

Date Published: July-25-2011

Trademarks

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are generally provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For technical support for products purchased directly from AudioCodes, or for customers subscribed to AudioCodes Customer Technical Support (ACTS), contact support@audiocodes.com.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Reader's Notes

1 Introduction

The Debug Recording (DR) mechanism duplicates all messages that are sent and/or received by the device and sends them to an external IP address. It is used for advanced debugging when it is required to analyze internal messages and signals. In addition, DR is useful for recording network traffic in environments in which hub / port mirroring isn't available and to record internal traffic between two endpoints on the same gateway.

DR can be used to capture the following message types:

- Digital signal processor (DSP) recording (see Section 3.1 on page 11):
  - RTP/RTCP streams that are sent and/or received by the device.
  - The actual voice signal (pulse-code modulation / PCM) that:
    - Arrives from the PSTN/PBX, before it enters the DSP.
    - Sent from the DSP to the PSTN/PBX.
  - Other internal information (such as DSP events and commands).
- PSTN trace - received/transmitted ISDN, CAS, and SS7 messages (see Section 3.2 on page 13).
- Control messages - SIP, MGCP, MEGACO, and TPNCP (see Section 3.3 on page 16).
- Networking streams (such as T.38, HTTP and SCTP (see Section 3.4 on page 16).

Notes:

- DSP, PSTN, Control and IP recording can be performed simultaneously.
- All DR rules that are defined through the CLI are deleted after the device is reset.
- DR can be used on a "live" device for debugging. For normal operation (unless otherwise requested by AudioCodes' support), DR must be disabled.
- DR doesn't require DSP channels and therefore, can be used when the device operates at full capacity.

Reader's Notes

2 Collecting DR Messages in Wireshark

The client that is used to capture the DR packets is the open source Wireshark program. This program can be downloaded from www.wireshark.org. In addition, AudioCodes proprietary plugin files (supplied in the software kit) must be located in the *plugin* folder of the installed Wireshark version program (typically, *C:\Program Files\WireShark\plugins\<Wireshark version>*).

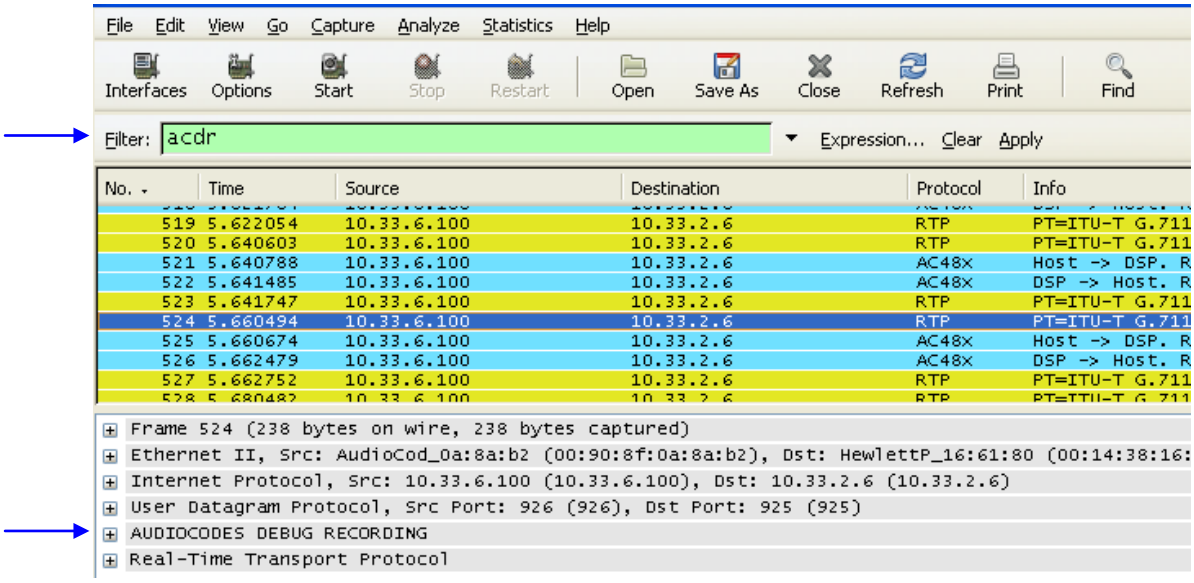
The default DR port is 925. This can be changed in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **ACDR**). When loaded, the Wireshark plugin dissects all packets on port 925 as DR packets.

Notes:

- The plugins for DR are per major software release.
- The plugins are applicable to Wireshark Version 99.08.
- The plugins are backward compatible.
- From Wireshark version 99.08, the **tpncp.dat** file must be located in the folder *C:\Program Files\WireShark\tpncp*.

Use the **acdr** filter to view the DR messages in Wireshark. The source IP address of the DR messages is always the OAMP IP address of the device. The DR mechanism adds to each message the proprietary header, "AUDIOCODES DEBUG RECORDING".

Figure 2-1: Viewing DR Messages in Wireshark



Reader's Notes

3 Debug Recording Modes

This section describes the different DR modes of operation.

3.1 DSP Recording

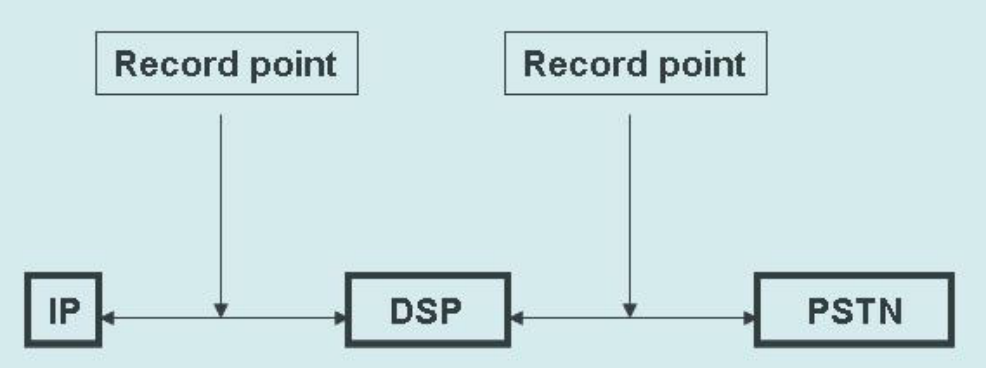
DSP recording should be used for analyzing voice-related issues such as: poor voice quality, echo, and fax / modem transmission.

The following messages are recorded **per channel**:

- Internal DSP packets and events.
- Incoming and outgoing RTP / RTCP streams (in the actual voice coder that is used by the channel).
- PCM - voice signal that arrives from and sent to the TDM (before it enters the DSP). The two streams are sent in G.711 A-law.

Notes: DSP recording cannot be used to record T.38 messages. To record T.38 messages, use the **AddIPTrafficTrace** command (see Section 3.4 on page 16)

Figure 3-1: DSP Record Points



When DSP recording is performed, for each call there are four media streams:

- **Network to DSP:** RTP messages received by the device
- **DSP to Network:** RTP messages sent by the device
- **TDM to DSP:** Voice signals received by the device from the PSTN/PBX
- **DSP to TDM:** Voice signals sent to the PBX/PSTN

Below is an example of the **AUDIOCODES DEBUG RECORDING** header that is used in DSP recording:

```
AUDIOCODES DEBUG RECORDING
    Version: 0x01
    Time Stamp: 0000EA32C298(3929.195160 sec)
    Source ID: 0
    Dest ID: 0
    Reserved: AA
    Trace Point: Dsp -> Network (1)
    Media Type: RTP Packet (1)
    Payload offset: 9
    Header Extension
        Packet destination IP address: 10.33.6.100 (10.33.6.100)
        Packet destination UDP port: 6010
        Packet source UDP port: 6000
        IP type of service: 184
```

3.1.1 DSP Recording for Mediant 800 MSBG

DSP recording for Mediant 800 includes different trace points. The voice path is:

Network <-> **VOIP encoder\decoder** <-> **DSP encoder\decoder** <-> **TDM**

Therefore, DSP recording for Mediant 800 provides four additional trace points. However, as they provide some common information, the recording produces two main traces:

- From the network, there are trace points “before VOIP decoder” (trace point #20) and “before DSP decoder” (18) that currently represent the same stream.
- From the Tel side, there are trace points “before VOIP encoder” (21) and “before NET encoder” (22).

These four trace points replace the DSP to network, and network to DSP.

3.1.2 Activating DSP Recording

You can activate DSP recording by using one of the following command options:

- **AddNextCallTrace:** Records the next *x* number of media calls.
- **AddTrunkBchannelTrace:** Records media calls according to trunk and B-channel (applicable only to digital PSTN interfaces).
- **AddChannelIdTrace:** Records media calls according to Channel ID (CID).

The **AddNextCallTrace** command is the most useful one to perform DSP recording when there are limited number of calls on the device. It cannot be used, for example, when a specific call needs to be recorded on a device that handles many dozens of calls. In this case, you need to isolate the problematic call on a specific Trunk/B-channel and use the **AddTrunkBchannelTrace** or **AddChannelIdTrace** commands.

3.2 PSTN Traces

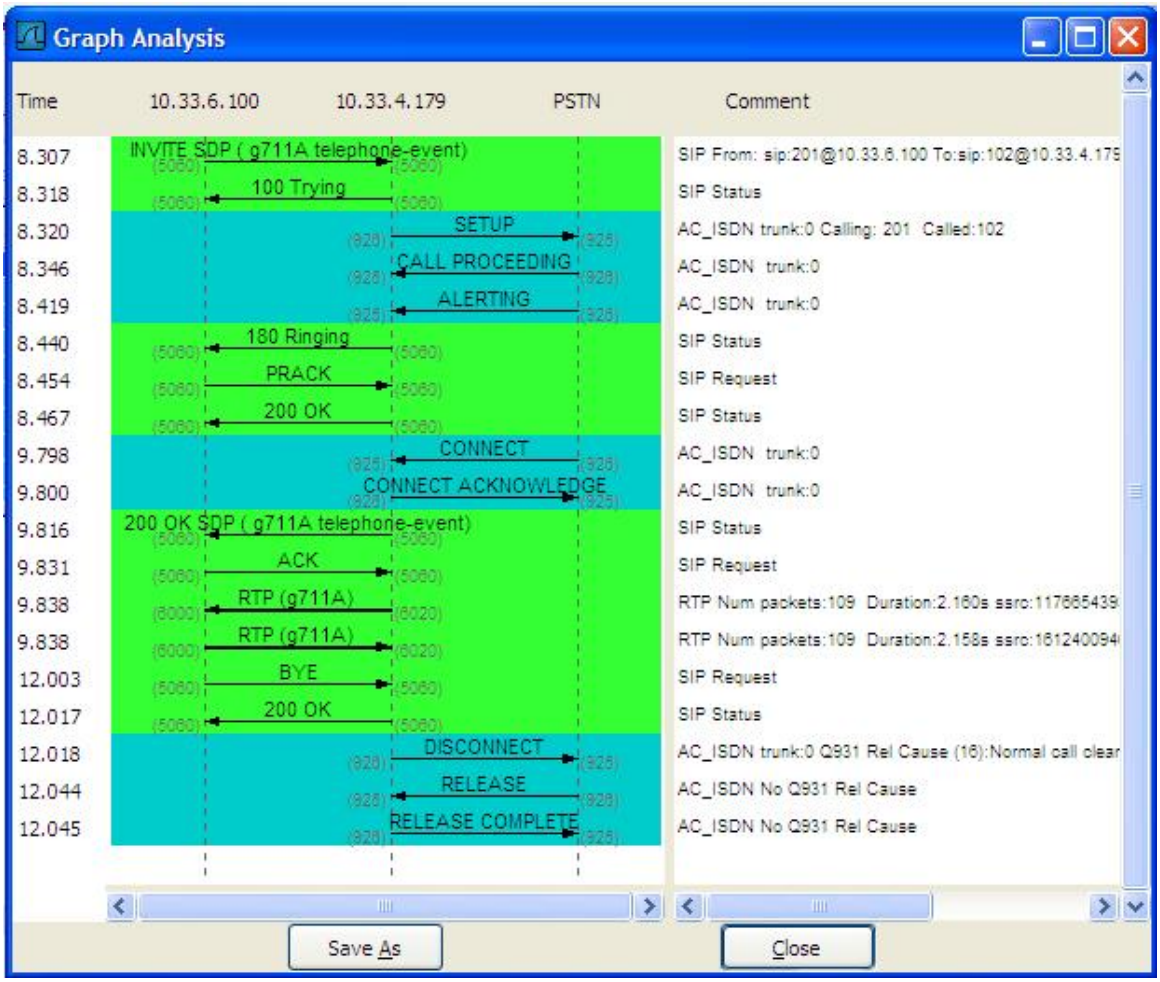
PSTN traces record ISDN, CAS and SS7 messages that are exchanged between the device and the PSTN/PBX switch.

Notes: PSTN trace messages can be sent through debug recording or directly to the Syslog. To send PSTN messages to the Syslog server, add the following parameter setting **PSTNReserved3=8** to the *ini* file, and then reset the device.

3.2.1 ISDN Traces

ISDN call flow can be viewed using the Wireshark's **q931** filter. Wireshark also allows you to convert the trace to a graph (**Statistics** > **VoIP Calls**) and viewed along with SIP messages.

Figure 3-2: ISDN Trace Graph in Wireshark



For ISDN messages, the additional header **NetBricks Trace** is added below **AUDIOCODES DEBUG RECORDING**. In addition, the protocol type is depicted by "SAPI: 1" (i.e., the "1" represents the Euro ISDN protocol). Below displays an example of such an ISDN trace.

```
AUDIOCODES DEBUG RECORDING
NetBricks Trace
System time: 3559
     Direction: Message received from internal server queue (73)
From (Entity origination ID): DL_D (DL LAPD Q.921) (100)
     To (Entity destination ID): PH_D (D channel physical) (68)
     Primitive code: 67
     NAI (Network Access ID): 0 -> number of trunk
     SAPI: 1
     Connection ID: 0
     Congestion flag: 0
     Allocated message: 2
     Allocated buffer: 3
     Allocated timer cell: 141
     IT Message stack counter: 120
     IT Buffer stack counter: 120
     Message congestion counter: 0
     Buffer congestion counter: 0
     IT Stack message congestion counter: 0
     IT Stack buffer congestion counter: 0
     Pointer to message: 689
     Pointer to buffer: 0
     Data size: 33
     Link Access Procedure, Channel D (LAPD)
     Q.931
        Protocol discriminator: Q.931
        Call reference value length: 2
        Call reference flag: Message sent from originating side
        Call reference value: 0300 - > can be used as a filter
to  identify entire ISDN call
        Message type: SETUP (0x05)
        Bearer capability
        Channel identification
        Calling party number: '201'
        Called party number: '102'
        Sending complete
```

3.2.2 CAS Traces

CAS traces include the **CAS Trace** header. The example below shows an example of a CAS trace:

```
AUDIOCODES DEBUG RECORDING
CAS Trace
     Timer: 1145504439
     From: DSP (0)
     Current State: 7
     Event: EV_DIAL_ENDED (15)
     Next State: -1
     Function Use: Unknown (-1)
        Parameter 1: -1
    Parameter 2: -1
     Parameter 3: -1
     Trunk Number: 3
     BChannel Number: 23
     Call Handle: 0
```

3.2.3 Enabling PSTN Traces per Trunk

By default, PSTN traces are disabled. You can enable PSTN traces for a specific trunk, using the device's Web interface, CLI, or *ini* file.

3.2.3.1 Using the Web Interface

The procedure below describes how to enable PSTN traces per trunk, using the Web interface.

➢ **To enable PSTN traces for a specific trunk, using the Web interface:**

1. Access the 'Trunk Settings' page (**Configuration** tab > **PSTN Settings** menu > **Trunk Settings**).
2. Stop the trunk, by clicking the **Stop Trunk** button.
3. Click the **Advanced Parameter List** link (located on the top-right corner of the page).
4. From the 'Trace Level' drop-down list, select "Full ISDN Trace".
5. Click the **Apply Trunk Settings** button.

3.2.3.2 Using the CLI

The procedure below describes how to enable PSTN traces per trunk, using the CLI.

➢ **To enable PSTN traces for a specific trunk, using the CLI:**

1. Access the CLI.
2. Enter the following commands:

```
pstn
PstnCOmmon
PstnSetTraceLevel <Trunk ID> -1 1
```

For example, to enable PSTN traces on the first Trunk, type **PstnSetTraceLevel 0 -1 1**

Note: For PSTN traces using CLI, you do not need to stop the trunk.

3.2.3.3 Using the INI File

The procedure below describes how to enable PSTN traces per trunk, using the *ini* file.

➢ **To enable PSTN traces for a specific trunk, using the ini file:**

- Load an *ini* file to the device with the following *ini* file parameter settings:

```
TraceLevel_0 = 1
; Where 0 depicts the first trunk number.
```

3.3 Control Traces

Control traces are used to record incoming and/or outgoing control messages (i.e., SIP, MEGACO, MGCP or TPNCP).

3.4 Network Traces

Network traces are used to record any IP stream that isn't associated with media (RTP/RTCP) according to destination and/or source IP address or port and Layer-4 protocol (UDP, TCP, SCTP or any other IP type as defined by http://www.iana.com). Network traces are typically used to record T.38, SCTP or HTTP.

Notes:

- To record T.38 messages, use the following commands:
  - **AddIPTrafficTrace host2Net udp a a all all**
  - **AddIPTrafficTrace Net2host udp a a all all**
- Network traces are not applicable to Mediant 800 MSBG and Mediant 1000 MSBG.

4 Fast Track for DR

Debug Recording activation is performed using the CLI interface under the *DebugRecording* directory. This section describes the basic procedures for quickly activating DR and collecting call traces.

For a more detailed description of all the DR commands, see Section 0 on page 20.

4.1 Activating DR

The procedure below describes how to initially activate the DR. Once activated, you can perform the required traces or recordings, as described in the subsequent subsections.

➢ **To activate the DR:**

1. Start a CLI management session.
2. At the prompt, type the following command to access the DebugRecording directory:
   ```
   DR
   ```
3. At the prompt, type the following command to terminate all active recordings, if any:
   ```
   STOP
   ```
4. At the prompt, type the following command to remove all previous recording rules:
   ```
   RTR ALL
   ```
5. At the prompt, type the following command to remove all DR targets (i.e., client IP addresses) from the list:
   ```
   RT ALL
   ```
6. At the prompt, type the following command to define the IP address of the PC (running Wireshark) to where the device sends its debug packets:
   ```
   AIT <IP address of the target>
   ```
7. Continue with the procedures described below according to the required recording.

4.2 Generating DSP Traces

The procedure below describes how to generate DSP traces.

➢ **To perform DSP traces:**

1. Setup DR, as described in Section 4.1 on page 17.
2. At the prompt, type the following command so that the next call on the device is recorded:
   ```
   ANCT ALL-WITH-PCM 1
   ```
3. At the prompt, type the following:
   ```
   START
   ```

4.3 Generating ISDN Traces

The procedure below describes how to generate ISDN traces.

➢ **To perform ISDN trace:**

1. Setup DR, as described in Section 4.1 on page 17.
2. Enable PSTN traces for a specific trunk (see Section 3.2.3 on page 15).
3. At the prompt, type the following command:

```
APST ISDN
```

4. At the prompt, type the following command:

```
START
```

4.4 Capturing Control Packets

The procedure below describes how to capture control (i.e., SIP, MGCP, MEGACO, or TPNCP) packets.

➢ **To capture Control packets**

1. Setup DR, as described in Section 4.1 on page 17.
2. At the prompt, type the following command (or MGCP/MEGACO/TPNCP):

```
AddIPControlTrace n2h SIP
```

   Note that instead of SIP, you can type **MGCP**, **MEGACO**, or **TPNCP**.

3. At the prompt, type the following command:

```
START
```

4.5 Capturing T.38 Traffic

The procedure below describes how to capture T.38 traffic.

➢ **To capture T.38 traffic you need to record all UDP messages that are sent received by the device:**

1. Setup DR, as described in Section 4.1 on page 17.
2. At the prompt, type the following command:

```
AddIPTrafficTrace host2Net udp a a all all
```

3. At the prompt, type the following command:

```
AddIPTrafficTrace Net2host udp a a all all
```

4. At the prompt, type the following command:

```
 START
```

4.6 Recording SCTP Traffic

The procedure below describes how to record Stream Control Transmission Protocol (SCTP) traffic.

➢ **To record SCTP traffic:**

1. Setup DR, as described in Section 4.1 on page 17.
2. At the prompt, type the following commands:

```
aiptt n2h 132 a a
aiptt h2n 132 a a
```

3. At the prompt, type the following command:

```
START
```

Reader's Notes

5 DR Command Reference

The tables below describe the DR commands. You can also view the description of a DR command in the CLI interface by simply typing the command name without any arguments.

Table 5-1: Client Setup Commands

Command	Parameters	Description
AddIpTarget	IPAddr [UDPPort]	Adds a Wireshark DR IP client to the list. UDPPort (optional): port on which to send the recorded packets (default is 925).
RemoveTarget	Index	Removes a DR client from the list. Index: index for the removed target (as displayed via ListTargets).
ListTargets		Displays the client list.
SetDefaultTarget	Index	Changes the default target. The default target is the first target added (AddTarget). Index: index for the default target (as displayed via ListTargets).

Table 5-2: Trace Rule Commands

Command	Parameters	Description
AddIPTrafficTrace	TracePoint PDUType SourcePort DestPort [SourceIP] [DestIP] [DebugTarget]	Record IP traffic. ▪ **Trace Point:** ✓ Net2Host = Inbound non-media traffic. ✓ Host2Net = outbound non-media traffic. ▪ **PDUType:** ✓ UDP = UDP traffic. ✓ TCP = TCP traffic. ✓ ICMP = ICMP traffic. ✓ IPType = Any other IP type (as defined by http://www.iana.com). ✓ A = All traffic types. ▪ **SourcePort:** Datagram's source port number (ALL for IP wildcard). ▪ **DestPort:** Datagram's destination port number (ALL for IP wildcard). ▪ **SourceIP (optional):** Datagram's source IP address (ALL for IP wildcard). ▪ **DestIP (optional):** Datagram's source IP address (ALL for IP wildcard). ▪ **DebugTarget (optional):** Debug target list index; if not specified, the default target is used.

Command	Parameters	Description
AddIPControlTrace	TracePoint ControlType [DebugTarget]	Records an IP control. ▪ **Trace Point:** ✓ Net2Host = Inbound/Outbound non-media traffic. ▪ **ControlType:** ✓ SIP = SIP traffic. ▪ **DebugTarget (optional):** Debug target list index; if not specified, the default target is used.
AddPstnSignalingTrace	PacketType [DebugTarget]	Records PSTN signaling. ▪ **Packet Type:** ✓ CAS = CAS signaling. ✓ ISDN = ISDN signaling. ✓ SS7 = SS7 signaling. ▪ **DebugTarget (optional):** Debug target list index; if not specified, the default target is used. **Notes:** ▪ Applicable only to Digital PSTN devices. ▪ To record PSTN signaling, 'PSTN Trace Level' (TraceLevel ini file) must be set to 1.
AddNextCallTrace	PacketType NumOfCalls [TraceType] [DebugTarget]	Records the next media calls. ▪ **Packet Type:** ✓ ALL = All media-related (internal DSP packets, RTP, RTCP, T38, events) of a specific call. ✓ ALL-WITH-PCM = All media-related and PCM traffic of a specific call. ▪ **NumOfCalls:** Amount of next media calls to record. (**Note:** Currently, only one call can be recorded.) ▪ **Trace Type (optional):** ✓ New (default) = Next new NumOfCalls calls to record. When these calls end, new calls are not recorded. ✓ Dynamic = Next new NumOfCalls calls to record. When these calls end, new calls are recorded until this trace is deleted. ▪ **RemoteIPAddr:** Captures number (according to the 'NumOfCalls' parameter) of next call, but with the special condition that these next calls should use only the specified remote IP address. For example: "AddNextCallTrace All 10 Dynamic 10.31.2.85" In this example, the next 10 dynamic RTP calls that activate the RTP to a specific remote IP address (i.e., 10.31.2.85) are recorded.

Command	Parameters	Description
AddTrunkBchannelTrace	PacketType TRUNK [TO_TRUNK] [BCHANNEL] [TO_BCHANNEL][DebugTarget]	Records media calls according to trunk and B-channel. ▪ **Packet Type:** ✓ ALL = All media-related (internal DSP packets, RTP, RTCP, T38, events) of a specific call. ✓ ALL-WITH-PCM = All media-related and PCM traffic of a specific call. ▪ **Trunk:** Start of range trunk number for recording. (Note: Currently, only 1 channel can be recorded.) ▪ **To_Trunk (optional):** End of range trunk number. ▪ **BChannel (optional):** Start of range B-Channel number for recording. ▪ **To_BChannel (optional):** End of range B-Channel number for recording. ▪ **DebugTarget (optional):** Debug target list index; if not specified, the default target is used. **Note:** Applicable only to Digital PSTN devices.
AddChannelIdTrace	PacketType Channel-Id [To Channel-Id][DebugTarget]	Records media calls according to CID. ▪ **Packet Type:** ✓ ALL = All media-related (internal DSP packets, RTP, RTCP, T38, events) of a specific call. ✓ ALL-WITH-PCM = All media-related and PCM traffic of a specific call. ▪ **Channel-Id:** Start of range channel ID number for recording. (**Note:** Currently, only one channel can be recorded for digital devices.) ✓ To Channel-Id (optional) = End of range channel ID number for recording. ▪ **DebugTarget (optional):** Debug target list index; if not specified, the default target is used.
RemoveTraceRule	Index	Removes TraceRule from list. Index: Rule index (as displayed via ListTraceRules). ALL for rule wildcard.
ListTraceRules	-	Displays the following: ▪ Added TraceRules. ▪ Status of Debug Recording tool (Active or Inactive). ▪ Number of debug recording connections.

Table 5-3: DR Activation Commands

Command	Parameters	Description
STARTrecording	-	Enables recording.
STOPrecording	-	Disables recording.

6 Collecting DR Messages from Device Startup

In some cases, especially for PSTN or SCTP, debug recording upon device startup (reset) is required. The procedure below describes how to enable TDM recording and to send the Syslog to Wireshark from the time the device resets.

➢ **To collect DR messages upon device startup (example):**

1. In the *ini* file, configure the following:

```
initialshellcommand = 'AdvancedMode;DR;AddIpTarget
10.33.2.29;AddNextCallTrace ALL-WITH-PCM 10;AddIPTrafficTrace
Host2Net A A 514 All All;start'
```

2. Load the *ini* file to the device.
3. Reset the device; the setting burns on the device's flash memory.

Configuration Note
Debug Recording

www.audiocodes.com