61210916L1-29.1D
July 2008

Configuration Guide

Voice Traffic over SIP Trunks

This configuration guide explains the concepts behind transmitting Voice over Internet Protocol (VoIP) over Session Initiation Protocol (SIP) trunks, using the ADTRAN voice products.

ADTRAN voice products combine data and voice into a single platform controlled by the ADTRAN Operating System (AOS). For detailed information regarding specific command syntax, refer to the *AOS Command Reference Guide* available on the *AOS Documentation* CD shipped with your AOS unit or on the Web at www.adtran.com.

This guide consists of the following sections:

VoIP Overview

VoIP provides cost savings to users by routing telephone calls over an Internet connection. VoIP converts analog voice into digital signals before they are sent over an Internet connection, and converts them back into analog voice signals on the destination side. This process is performed by Pulse Code Modulation (PCM - CODEC G.711) which requires 64 kbps of bandwidth or by the Analysis-by-Synthesis (AbS - CODEC G.729) hybrid which only requires 8 kbps of bandwidth. Both CODECs provide high-quality sound and voice coding and are supported in AOS voice products.

AOS uses SIP for VoIP applications, providing interoperability with industry-leading softswitches, feature servers, and gateways. AOS devices convert SIP signaling into ordinary Time Division Multiplex (TDM) analog and digital voice service, acting as the SIP gateway. This functionality allows AOS voice products to deliver voice services to both IP phones and traditional telephony equipment simultaneously.

Although SIP is a fairly new IP signaling protocol, its simplicity and interoperability have made it very desirable. The Internet Engineering Task Force (IETF) has officially adopted SIP as the standard VoIP application layer protocol.

Figure 1 shows a network (Internet) application where the SIP trunk carries voice and data over a T1. Voice is multiplexed from the POTS or DSX-1 ports, and data is routed to the LAN.

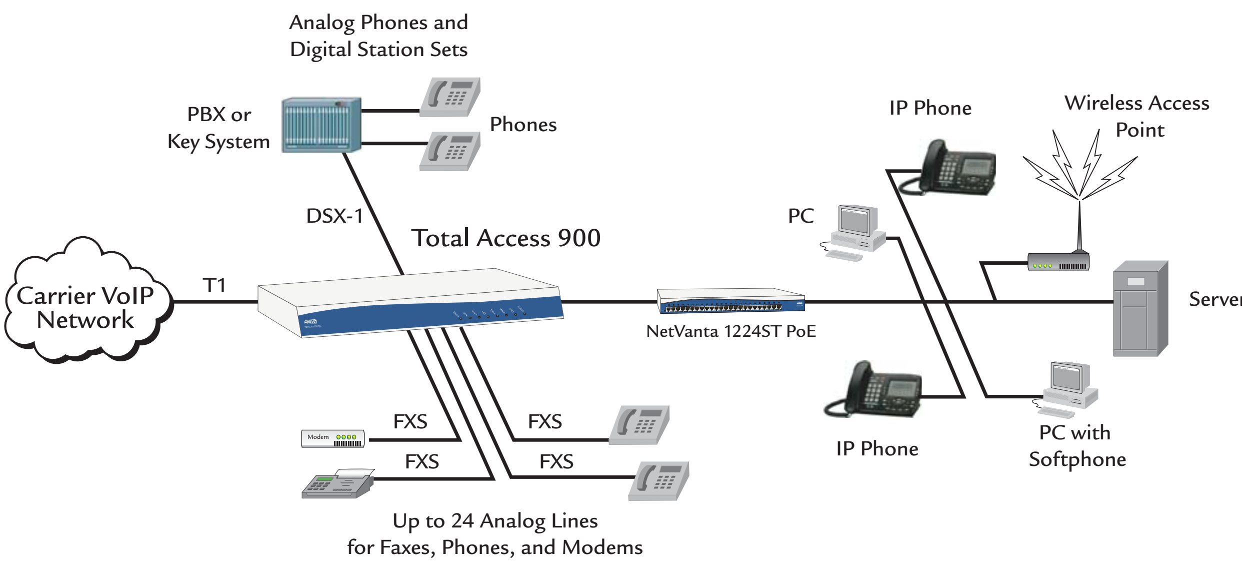

Figure 1. AOS Voice Products VoIP Application

SIP Overview

SIP is a client/server signaling protocol responsible for the initiation and management of IP voice communication sessions for AOS voice products. SIP is designed to control call setup and tear down between VoIP endpoint devices. The most common use of SIP in Internet technologies is for services, such as instant messaging, IP voice calls, video Web cams, and IP Centrex service. The basic function of SIP is to locate endpoints, signal a desire to communicate, establish sessions, and tear down sessions between endpoints. The current version of SIP (2.0) is defined in RFC 3261. The protocol responsible for transporting the voice traffic over the SIP connections is Realtime Transport Protocol (RTP). To learn more about RTP, refer to the *Voice Traffic over SIP Trunks* section below.

Understanding SIP Network Devices

SIP client endpoint devices, such as IP phones, are referred to as user agents (UAs). The IP phone UA functions as the UA client (UAC) that originates calls, the UA server (UAS) that listens for incoming calls, or as both the UAC and the UAS. The UAC sends SIP requests and receives SIP responses. UAS generates a response message to a SIP request, which accepts, rejects, or redirects the UAC's request.

The SIP registrar server accepts register requests from UAs. Clients register with the SIP registrar server each time a connected SIP device is powered up. The SIP registration process makes it possible for UAs to be identified by the system, as well as calls to be directed to the proper UA.

The SIP proxy server also acts as both server and client in the VoIP network by routing calls to the receiving UA devices. The SIP proxy server's main duties are to act as intermediary, making requests for clients on the behalf of other clients and routing requests to other intermediaries closer to the targeted user.

Voice Traffic over SIP Trunks

SIP client/server properties are very similar to email accounts and users. SIP uses a Uniform Resource Identifier (URI) comparable to an email address. The SIP URI format is SIP Number@SIP Service Domain. For example, the SIP number can be alphanumeric like maryjo@VoIP-provider.com or 5551234@VoIP-provider.com. The service domain for the VoIP provider is the name used in the SIP URI (VoIP-provider.com). Often, an IP address is used in place of a domain name. The SIP client sends the VoIP call request, and the SIP server responds. Again, SIP client and server can be a single device, allowing users to make and receive VoIP calls.

SIP uses the Session Description Protocol (SDP) to format the SIP message body in order to negotiate an RTP connection between two or more UAs. The ports used for this connection will always be selected in a pair, with the even port used for RTP. Remember, RTP has the important role of actually transporting the SIP voice traffic.

Hardware and Software Requirements and Limitations

The SIP trunk features were introduced in AOS version 11.01.00 and are currently available in the following products:

- Total Access 900 Series
- Total Access 900e Series
- NetVanta 6355
- NetVanta 7000 Series

AOS firmware version 14.01 or later is required on your system in order to use the multiple SIP trunk features.

Voice Configuration

NOTE	*Default router configuration is loaded in the unit prior to factory release. Confirm that your router's Ethernet interfaces and IP addresses are configured properly. Standard router configuration must be done before continuing with the SIP voice configuration.*

NOTE	*The configuration parameters used in the example are for instructional purposes only. Please replace all underlined entries (**example**) with your specific parameters to configure your application. For detailed information on specific commands, refer to the **AOS Command Reference Guide** provided on the AOS Documentation CD shipped with your AOS unit or available on the Web at www.adtran.com.*

Enabling SIP Functionality and Basic Setup

When configuring VoIP applications, RTP traffic requires an associated IP address in order to know which interface to use for voice transport. Use the **media-gateway ip** command to associate an IP address source to use for RTP traffic.

Step 1: Enable SIP

At the (config)**#** prompt, enter **ip sip**. The default value for SIP signaling QoS DSCP is **26**. Use the **ip sip qos dscp** <*DSCP values*> command to change the setting. The default value for QoS DSCP RTP (media) packets is **46**. Use the **ip rtp qos dscp** <*DSCP value*> command to change the setting.

Example

(config)#**ip sip**

Step 2: Verify the media gateway configuration

This step is very important because it specifies which interface and IP address to use for RTP traffic. The following example specifies that RTP traffic leave the unit on the PPP interface (**ppp 1**) with the media gateway IP address set to **primary**. Static, DHCP, or negotiated addresses are also valid.

Example

(config)#**interface ppp 1**
(config-ppp 1)#**ip address 10.10.10.1 255.255.255.252**
(config-ppp 1)#**media-gateway ip primary**
(config-ppp 1)#**no shutdown**
(config-ppp 1)#**exit**

> NOTE: *Additional* ***media-gateway*** *subcommands are available. Enter* ***media-gateway loopback*** *command to use an IP address statically defined to a loopback interface. Enter* ***media-gateway ip secondary*** *command to use the statically defined secondary IP address of this interface to be used for RTP traffic.*

Step 3: Verify the dial plan configuration

The dial plan listens for patterns of digits entered by the user in order to properly route calls, as well as allow the system to recognize dialed numbers as a particular type of call. Configure the dial plan patterns if they have not been previously defined.

> NOTE: *For more information on configuring dial plans, refer to the* ***Switchboard and Dial Plan Configuration Guide****, available on the AOS Documentation CD shipped with your AOS unit or on the Web at www.adtran.com.*

Step 4: Create a CODEC group to be used for the voice trunk

The first CODEC type entered will automatically become the primary CODEC for this group. The option to make this CODEC group the default for all new users can be implemented by entering the **default** command within the CODEC list configuration.

Command Syntax

(config)#**voice codec-list** <*name*>

<*name*> Identifies the CODEC group for call negotiation with the SIP server.

(config-codec)#**codec** <*primary codec*>

<*primary codec*> Specifies the primary CODEC type for this application.

(config-codec)#**codec** <*secondary codec*>

<*secondary codec*> Specifies the secondary CODEC type for this application.

(config-codec)#**default**

Optional. Specifies that this CODEC list be used as the default for the system.

(config-codec)#**exit**

Example Configuration

(config)#**voice codec-list** **<u>Trunk</u>**
(config-codec)#**codec** **<u>g729</u>**
(config-codec)#**codec** **<u>g711ulaw</u>**
(config-codec)#**default**
(config-codec)#**exit**

SIP Trunk Configuration

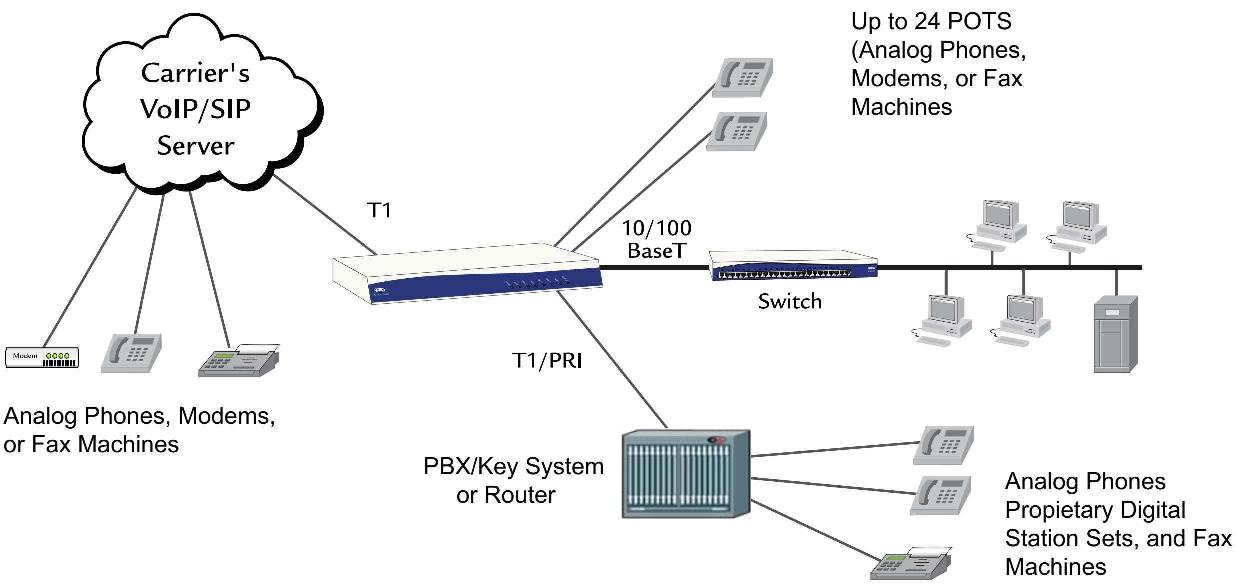

Figure 2. SIP Trunk Application

Configuring a Single SIP Trunk

Step 1: Create a SIP trunk account and link the SIP servers

The SIP servers' configuration specifies to which trunk the SIP messages are sent for this connection. The default IP address is **0.0.0.0** and the default TCP/UDP port is **5060** for voice trunks.

Command Syntax

(config)#**voice trunk** <*Txx*> **type sip**

<*Txx*> Specifies the trunk identity in the format Txx, where xx is the trunk ID number between 01 and 99 (for example, **T01**).

(config-Txx)#**sip-server primary** <*FQDN or IP address*> **[tcp | udp]** <*number*>

<FQDN or IP address> Specifies the fully qualified domain name (FQDN) or IP address of the SIP server.
<number> Optional. Specifies a custom TCP or UDP destination port number. The default TCP/UDP port is 5060.

(config-Txx)#**registrar primary** *<FQDN or IP address>* **[tcp | udp]** *<number>*

<FQDN or IP address> Specifies the FQDN or IP address of the SIP server.
<number> Optional. Specifies a custom TCP or UDP destination port number. The default TCP/UDP port is 5060.

(config-Txx)#**outbound proxy primary** *<FQDN or IP address>* **[tcp | udp]** *<number>*

<FQDN or IP address> Specifies the FQDN or IP address of the SIP server.
<number> Optional. Specifies a custom TCP or UDP destination port number. The default TCP/UDP port is 5060.

(config-Txx)#**codec-group** *<name>*

<name> Applies a previously created CODEC group.

Using the following range option reduces the time required to program individual user registration and authentication settings:

(config-Txx)#**register** *<name>*

or

(config-Txx)#**register range** *<number> <number>*

or

(config-Txx)#**register** *<name>* **auth-name** *<username>* **password** *<word>*

or

(config-Txx)#**register range** *<begin> <end>*

or

(config-Txx)#**register range** *<begin> <end>* **auth-name** *<username>* **password** *<word>*

or

(config-Txx)#**register range** *<begin> <end>* **auth-name range** *<begin> <end>* **password range** *<begin> <end>*

or

(config-Txx)#**register range** *<begin> <end>* **auth-name range** *<begin> <end>* **password** *<word>*

<name> Specifies the name of the SIP trunk to register.
<number> Optional. Registers the specified number or range of numbers from the trunk to the SIP server. This is primarily required in DSX-1 (PBX) applications when the SIP server needs to receive registration information from the unit.

range <*begin*> <*end*> Specifies the beginning and ending of the range to register SIP users, authentication names, and/or passwords.
auth-name <*username*> Optional. Specifies the user name for authentication.
password <*word*> Optional. Specifies the password for authentication.

(config-Txx)#**authentication username** <*username*> **password** <*password*>

<*username*> Specifies the user name to be used for registration.
<*password*> Specifies the password to be used for registration.

Each port that registers with the SIP server will use the defined user name and password. If all users on the trunk use the same user name/password, enter the user name and password for authentication under the trunk. Otherwise, enter authentication information for each user individually in the Voice User command set which overrides the setting of this command. For more information on the Voice User command set, refer to the *AOS Command Reference Guide* available on the *AOS Documentation* CD shipped with your AOS unit or on the Web at www.adtran.com.

(config-Txx)#**match** <*number*> **substitute** <*number*>

<*number*> Specifies the number to match and the number to send in its place.

Optional. Substitutes a different number for the number originally dialed by a user of the system. For example, matching 411 and substituting 555-1212 would change the outgoing number to 555-1212.

This command is useful because it allows users to dial only 7 digits when 10-digit dialing is required. For example, **match NXX-XXXX substitute 256-NXX-XXXX** appends an area code (256) to the 7-digit dial pattern. If no match occurs (or no match statements have been entered), the original dialed number will be propagated without being modified.

Example Configuration

(config)#**voice trunk <u>T01</u> type sip**
(config-T01)#**sip-server primary <u>172.16.100.2</u>**
(config-T01)#**registrar primary <u>172.16.100.2</u>**
(config-T01)#**outbound proxy primary <u>172.16.100.3</u>**
(config-T01)#**codec-group <u>Trunk</u>**
(config-T01)#**register range 5559300 5559399**
(config-T01)#**authentication username <u>hsvhq</u> password <u>adtran</u>**
(config-T01)#**match <u>0</u> substitute <u>6111</u>**
(config-T01)#**exit**

Step 2: Create the voice trunk group

Assign the existing trunk to the trunk group in order to direct outbound calls to the proper trunk. Also, define the dialed number patterns to be accepted or rejected. For more information about dialing patterns, refer to the *Switchboard and Dial Plan Configuration Guide* available on the *AOS Documentation* CD shipped with your AOS unit or on the Web at www.adtran.com. The **cost** command is used when multiple trunks can accept a certain number pattern. The lowest cost value determines the trunk to which the call is routed.

Command Syntax

(config)#**voice grouped-trunk** <*trunk id*>

<*trunk ID*> Specifies a name to identify this grouped trunk.

(config-TrunkGroup)#**trunk** <*Txx*>

<*Txx*> Specifies an existing trunk to add to this trunk group for outbound calls. Enter the trunk identifier in the form Txx, where xx is the trunk ID between 01 and 99 (for example, **T01**).

(config-TrunkGroup)#**accept** <*pattern*> **cost** <*value*>

<*pattern*> Specifies the dialed number patterns that are allowed on this trunk group. The **$** symbol is a wildcard that allows any number to pass.

<*value*> Optional. Specifies the cost value.

(config-TrunkGroup)#**reject** <*pattern*>

<*pattern*> Specifies the dialed number pattern that is not allowed on this trunk. Use this command to restrict certain types of outbound calls.

Example Configuration

(config)#**voice grouped-trunk** **<u>TrunkGroup</u>**
(config-TrunkGroup)#**trunk** **<u>T01</u>**
(config-TrunkGroup)#**accept** **<u>NXX-XXXX</u>** **cost** **<u>0</u>**
(config-TrunkGroup)#**accept** **<u>$</u>**
(config-TrunkGroup)#**reject** **<u>1-NXX-NXX-XXXX</u>**
(config)#**exit**
#**copy running-config startup-config**

Configuring Multiple SIP Trunks

Configuring multiple SIP trunks allows users to connect to several different VoIP SIP service providers for added flexibility and network redundancy. More flexibility means greater savings for most networks by using specific VoIP service providers for long distance and international calls. This allows the subscribers to avoid using public switched telephone networks (PSTNs), which typically cost more. With multiple SIP trunks, the added redundancy ensures network reliability in case of network failure. Multiple SIP trunk applications support up to three different SIP trunk connections. Refer to the examples below for different ways to configure this application.

Examples

Create the SIP trunks.

(config)#**voice trunk <u>T04</u> type sip**
(config-T04)#**sip-server primary <u>172.16.100.4</u>**
(config-T04)#**registrar primary <u>172.16.100.4</u>**
(config-T04)#**exit**
(config)#**voice user 5500**
(config-5500)#**sip-identity 4400 T04 register**
(config-5500)#**max-num-calls 20**
(config-5500)#**exit**

(config)#**voice trunk <u>T05</u> type sip**
(config-T05)#**sip-server primary <u>172.16.100.5</u>**
(config-T05)#**registrar primary <u>172.16.100.5</u>**
(config-T05)#**exit**
(config)#**voice user 5500**
(config-5500)#**sip-identity 5500 T05 register**
(config-5500)#**max-num-calls 20**
(config-5500)#**exit**

Group trunks to combine one or more trunk accounts.

(config)#**voice grouped-trunk <u>Local Calls</u>**
(config-TrunkGroup)#**trunk <u>T04</u>**
(config-TrunkGroup)#**accept <u>NXX-XXXX</u> cost <u>0</u>**
(config-TrunkGroup)#**accept <u>NXX-NXX-XXXX</u> cost <u>4</u>**
(config-TrunkGroup)#**reject <u>1-NXX-NXX-XXXX</u>**
(config)#**exit**
#**copy running-config startup-config**

(config)#**voice grouped-trunk <u>Long Distance</u>**
(config-TrunkGroup)#**trunk <u>T05</u>**
(config-TrunkGroup)#**accept <u>1-NXX-NXX-XXXX</u> cost <u>1</u>**
(config-TrunkGroup)#**accept <u>NXX-XXXX</u> cost <u>20</u>**
(config)#**exit**
#**copy running-config startup-config**

Or use this alternate method of multiple trunk group configuration.

(config)#**voice grouped-trunk <u>Long Distance</u>**
(config-TrunkGroup)#**trunk <u>T04</u>**
(config-TrunkGroup)#**trunk <u>T05</u>**
(config-TrunkGroup)#**accept <u>NXX-XXXX</u> cost <u>0</u>**
(config-TrunkGroup)#**accept <u>1$</u> cost <u>10</u>**
(config-TrunkGroup)#**accept <u>011$</u> cost <u>20</u>**
(config)#**exit**
#**copy running-config startup-config**

Configuration Command Summary

	Command	Description
Step 1	(config)#**ip sip**	Enter the Global configuration mode to enable SIP.
Step 2	(config)#**interface ppp 1**	Verify the media gateway configuration. This step is very important because it specifies which interface and IP address to use for RTP traffic.
	(config-ppp 1)#**ip address 10.10.10.1 255.255.255.252**	Specify the IP address.
	(config-ppp 1)#**cross-connect 1 t1** *<slot/port> <tdm group>* **ppp1**	Enter **cross-connect** command to connect DS0s assigned in the T1 network connection TDM group to the virtual PPP interface 1. The cross-connect "1" is a label. Refer to the *AOS Command Reference Guide* available on the AOS Documentation CD shipped with your AOS unit or on the Web at www.adtran.com for information on configuring the T1 interface, and cross connecting T1 interface with the virtual PPP interface.
	(config-ppp 1)#**media-gateway ip primary**	Specify the media gateway.
	(config-ppp 1)#**no shutdown**	Activate the PPP interface mode.
	(config-ppp 1)#**exit**	Exit the PPP configuration menu.
	(config)#**do show interface ppp1**	Confirm the interface has been activated.
Step 3	(config)#**do show voice dial-plan** *<template number>*	Verify the dial plan configuration.
Step 4	(config)#**voice codec-list** *<name>*	Create the CODEC list for the voice trunk(s).
	(config-codec)#**codec** *<primary codec>*	Specify the primary CODEC type for this application.
	(config-codec)#**codec** *<secondary codec>*	Specify the secondary CODEC type for the application.
	Optional (config-codec)#**default**	Specify the previously configured CODEC list to be the default for the system.
	(config-codec)#**exit**	Exit the CODEC configuration mode.

	Command	Description
Step 5	(config)#**voice trunk** <*Txx*> **type sip**	Specify the voice trunk identity and type.
	(config-Txx)#**sip-server primary** <*FQDN or IP address*> **[tcp | udp]** <*number*>	Specify the FQDN or IP address of the SIP server (primary). The custom TCP or UDP port setting is optional.
	(config-Txx)#**registrar primary** <*FQDN or IP address*> **[tcp | udp]** <*number*>	Specify the FQDN or IP address of the SIP server (registrar). The custom TCP or UDP port setting is optional.
	(config-Txx)#**outbound proxy primary** <*FQDN or IP address*> **[tcp | udp]** <*number*>	Specify the FQDN or IP address of the SIP server (outbound). The custom TCP or UDP port setting is optional.
	(config-Txx)#**codec-group** <*name*>	Apply the CODEC list for this trunk group.
	(config-Txx)#**register** <*number*> or (config-Txx)#**register range** <*number*> <*number*>	Optional. Registers the specified number or range of numbers from the trunk to the SIP server. This is primarily required in DSX-1 (PBX) applications when the SIP server needs to receive registration information from the unit.
	(config-Txx)#**authentication username** <*username*> **password** <*password*>	Specify the username and password. If all users on the trunk use the same user name/password, enter the user name and password for authentication under the trunk. Otherwise, enter authentication information for each user individually in the Voice User command, set which overrides the setting of this command.
	(config-Txx)#**match** <*number*> **substitute** <*number*>	Optional. Substitutes a different number for the number originally dialed by a user of the system. For example, match 411 to 555-1212 would change the outgoing number to 555-1212.
	(config-Txx)#**exit**	Exit the trunk configuration.
Step 6	(config)#**voice grouped-trunk** <*trunk ID*>	Create the voice trunk group. Assign the existing trunk to the trunk group in order to direct outbound calls to the proper trunk (T01).
	(config-TrunkGroup)#**trunk** <*Txx*>	Specify the trunk to add to the trunk group for outbound calls. Enter the trunk identifier in the form Txx, where xx is the trunk ID between 01 and 99 (for example, T01).

	Command	Description
	(config-TrunkGroup)#**accept** <*pattern*> **cost** <*value*>	Specify the dialed number patterns that are allowed on this trunk group. The **$** symbol used as a wildcard allows any number to pass.
	(config-TrunkGroup)#**reject** <*pattern*>	Specify the dialed number pattern(s) not allowed on this trunk. Use this command to restrict certain outbound calls.
	(config)#**exit**	Exit the Trunk Group configuration mode.
	#**copy running-config startup-config**	Save the configuration.

Troubleshooting

Once the SIP trunks are configured, several different commands can be issued from the Enable mode in the CLI to assist with troubleshooting or monitoring an issue. For a complete list of voice-related **debug** commands, refer to the *AOS Command Reference Guide* available on the *AOS Documentation* CD shipped with your AOS unit or on the Web at www.adtran.com.

> NOTE: *Turning on a large amount of debug information can adversely affect the performance of your unit.*

Table 1. Voice Troubleshooting Commands

Command	Explanation
debug sip trunk-registration [*<trunk>* **|** *<trunk> <trunk id>***]**	Activates SIP trunk-registration event debug messages. Specifying a particular trunk is optional. For example: Txx (T01), where *xx* is the trunk's two-digit identifier and *<trunk id>* is the specific name associated with the trunk.
debug voice phonemanager [*<extension>* **|** *<slot:port>***]**	Activates phone manager (PM) event debug messages for all extensions or for a specific extension (for SIP phones), or for a virtual interface (slot and port), such as SIP. Each voice user automatically has a PM created after initial configuration of the voice user.
debug voice stationaccount [*<extension>***]**	Activates station account (SA) event debug messages for a specific extension. The SA governs all voice user features and specific configurations. The SA communicates with the switchboard on behalf of its associated voice user.
debug voice trunkaccount [*<trunk id>* **|** *<trunk id> <appearance>***]**	Activates trunk account event debug messages for a specific extension.
debug voice trunkmanager [*<trunk id>* **|** *<trunk id> <appearance>***]**	Activates trunk manager event debug messages for a specific extension.
debug voice trunkport [*<slot:port:DSO>***]**	Activates trunk port event debug messages for a specific slot, port, and DSO.

Debug SIP Commands

Use the **debug sip** commands to activate debug messages associated with SIP events. Debug messages are displayed (in real time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages, or use **undebug all** to disable all debug messaging. Use the **show debugging** command to view which debug message command are activate.

debug sip trunk-registration

Variations of the **debug sip trunk-registration** command include the following:

debug sip trunk-registration
debug sip trunk-registration <*trunk*>
debug sip trunk-registration <*trunk*> <*trunk id*>

Debug Voice Commands

Use the **debug voice** commands to activate debug messages associated with voice functionality. Debug messages are displayed (in real time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages, or use **undebug all** to disable all debug messaging.

debug voice phonemanager and debug voice stationaccount

The **debug voice phonemanager** and **debug voice stationaccount** commands are helpful when troubleshooting issues with call connections. Use these commands to verify hook events, dialed number patterns, incoming calls ringing once they are properly routed, etc. The debug output displays dial tone generation, busy tone, etc.

When troubleshooting call failures, scan the debug messages for causes of incomplete connection. The PM and SA work together to manage individual voice users along with their assigned interfaces. The PM links the voice stack with the physical wire connection. It is responsible for line events, such as on-hook, off-hook, flash, etc. PM also signals state changes associated with the line events and passes this information to the SA. The SA receives the line events for processing. This information is passed between the PM and SA and vice versa. Eventually, the SA takes ownership for all switchboard entries for a given voice user. Once a call is accepted through the switchboard, the SA validates the call and passes necessary state changes back to the PM.

Variations of the **debug voice phonemanager** command include the following:

debug voice phonemanager
debug voice phonemanager <*extension*>
debug voice phonemanager <*slot:port*>

Variations of the **debug voice stationaccount** command include the following:

debug voice stationaccount
debug voice stationaccount <*extension*>

Usage Examples

The following usage examples activate debug messages associated with PM and SA voice functionality:

>**enable**
#**debug voice phonemanager 0:1**

```
06:46:47: PM0:1Idle                    Processed OFFHOOK
06:46:47: PM0:1State change            >> Idle->Requesting Dialtone
06:46:47: PM0:1Requesting Dialtone     CACHG:ReqDigits on primary CA
06:46:47: PM0:1State change            >> Requesting Dialtone->SendingDigits
06:46:49: PM0:1SendingDigits           Digit 9 processed
06:46:49: PM0:1SendingDigits           Digit 9 processed
06:46:49: PM0:1SendingDigits           Digit 9 processed
06:46:49: PM0:1SendingDigits           Digit 9 processed
06:46:50: PM0:1SendingDigits           Digit # processed
06:46:50: PM0:1State change            >> SendingDigits->Call Pending
06:46:50: PM0:1Call Pending            Acct->PM INFO, not used
06:46:50: PM0:1Call Pending            processed CACHG:Connected
06:46:50: PM0:1State change            >> Call Pending->Connected
06:46:53: PM0:1Connected               Processed ONHOOK
06:46:53: PM0:1State change            >> Connected->Idle
06:46:53: PM0:1Idle                    Dropped CACHG w/Call State not RINGING
```

The sample output above is a PM debug that shows the analog station on port 0:1 going off-hook, playing dialtone, and dialing 9999#. Next, it shows the call connection, but it is not aware of the connection process through the network. The call stays connected for three seconds before the user hangs up. Finally, the line state changes back to idle. Use debug PM to investigate a specific slot:port or extension suspected of causing issues.

>**enable**
#**debug voice stationaccount 2445**

```
06:50:54: SA9026Ca:0 Idle              rcvd: AcctPhoneMgr_appearance(ON) from PM
06:50:54: SA9026Ca:0 Idle              rcvd: AcctPhoneMgr_appearance(ON) from PM
06:50:54: SA9026Ca:0 State change      >> Idle->DigitGathering
06:50:54: SA9026Ca:0 DigitGathering    sent: AcctPhoneMgr_cachg(CAS_ReqDigits) to PM
06:50:55: SA9026Ca:0 DigitGathering    rcvd: AcctPhoneMgr_dialDigit(9) from PM
06:50:55: SA9026Ca:0 DigitGathering    rcvd: AcctPhoneMgr_dialDigit(9) from PM
06:50:55: SA9026Ca:0 DigitGathering    rcvd: AcctPhoneMgr_dialDigit(9) from PM
06:50:56: SA9026Ca:0 DigitGathering    rcvd: AcctPhoneMgr_dialDigit(9) from PM
06:50:56: SA9026Ca:0 DigitGathering    rcvd: AcctPhoneMgr_dialDigit(#) from PM
06:50:56: SA9026Ca:0 State change      >> DigitGathering->DigitGathering
06:50:56: SA9026Ca:0 DigitGathering    sent: AcctPhoneMgr_cachg(CAS_Active) to PM
06:50:56: SA9026Ca:0 DigitGathering    sent: call to SB
06:50:56: SA9026Ca:0 State change      >> DigitGathering->CallPending
06:50:56: SA9026Ca:0 CallPending       rcvd: callResponse from SB
06:50:56: SA9026Ca:0 State change      >> CallPending->Calling
06:50:56: SA9026Ca:0 Calling           rcvd: deliverResponse from SB
06:50:56: SA9026Ca:0 Calling           sent: AcctPhoneMgr_info to PM
06:50:56: SA9026Ca:0 Calling           rcvd: connect from SB
06:50:56: SA9026Ca:0 State change      >> Calling->ConnectPending
06:50:56: SA9026Ca:0 ConnectPending    sent: AcctPhoneMgr_cachg(CAS_Connected) to PM
06:50:56: SA9026Ca:0 ConnectPending    rcvd: AcctPhoneMgr_connect from PM
```


06:50:56: SA9026Ca:0 ConnectPending	sent: AcctPhoneMgr_faxModem to PM
06:50:56: SA9026Ca:0 ConnectPending	sent: connectResponse(pass) to SB
06:50:56: SA9026Ca:0 State change	>> ConnectPending->Connected
06:50:58: SA9026Ca:0 Connected	rcvd: AcctPhoneMgr_appearance(OFF) from PM
06:50:58: SA9026Ca:0 Connected	sent: clearCall to SB
06:50:58: SA9026Ca:0 State change	>> Connected->Clearing
06:50:58: SA9026Ca:0 Clearing	rcvd: clearResponse from SB
06:50:58: SA9026Ca:0 State change	>> Clearing->Idle
06:50:58: SA9026Ca:0 Idle	sent: AcctPhoneMgr_cachg(CAS_Idle) to PM

The sample output above is an SA debug that shows most of the information available with the PM debug, but with more detailed switchboard information. The SA receives an off-hook event from the PM as an incoming call appearance. The digit collection and state changes pass back and forth, as well as connect/clearing messages from the switchboard. When troubleshooting, SA debug messages give you the ability to specify a particular station's phone number, allowing only events of that station to be displayed. Use this troubleshooting tool to identify call failures or issues with a specific station.

debug voice trunkaccount

Variations of the **debug voice trunkaccount** command include the following:

debug voice trunkaccount
debug voice trunkaccount *<trunk id>*
debug voice trunkaccount *<trunk id> <appearance>*

Example Output

#**debug voice trunkaccount**

15:05:13 TA.T01 01 TAConnected	ClearCall event accepted
15:05:13 TA.T01 01 State change	>> TAConnected->TAClearingComplete
15:05:13 TA.T01 01 TAClearingComplete	Clear Local Variables
15:05:13 TA.T01 01 State change	>> TAClearingComplete->TAIdle
15:05:13 TA.T01 01 TAClearingComplete	Processing an appearance OFF
15:05:32 TA.T01 22 TAIdle	Deliver event accepted
15:05:32 TA.T01 22 State change	>> TAIdle->TAOutGoing
15:05:32 TA.T01 22 SUBSTITUTION:	M#8967 S#8967
15:05:32 TA.T01 ERROR!	preConnectResponse ignored
15:05:35 TA.T01 22 State change	>> TAOutGoing->TAConnectWaitOut
15:05:35 TA.T01 22 TAConnectWaitOut	ConnectResp event accepted
15:05:35 TA.T01 22 State change	>> TAConnectWaitOut->TAConnected
15:05:47 TA.T01 22 TAConnected	ClearCall event accepted
15:05:47 TA.T01 22 State change	>> TAConnected->TAClearingComplete
15:05:48 TA.T01 22 TAClearingComplete	Clear Local Variables
15:05:48 TA.T01 22 State change	>> TAClearingComplete->TAIdle
15:05:48 TA.T01 22 TAClearingComplete	Processing an appearance OFF

debug voice trunkmanager

Variations of the **debug voice trunkmanager** command include the following:

debug voice trunkmanager
debug voice trunkmanager *<trunk id>*
debug voice trunkmanager *<trunk id> <appearance>*

Example Output

#**debug voice trunkmanager**
15:17:53 TM.T01 23 Incoming Call Accepted
15:17:53 TM.T01 23 tachg_TAInboundCall do nothing
15:17:54 TM.T01 23 tachg_Alerting
15:18:17 TM.T01 23 tachg_Connected
15:18:59 TM.T01 23 tachg_Clearing
15:19:00 TM.T01 23 IsdnTrunkManager::CallRelease - release Channel.
15:19:00 TM.T01 23 tachg_Idle
15:19:02 TM.T01 19 tachg_Clearing
15:19:02 TM.T01 19 IsdnTrunkManager::CallRelease - release Channel.
15:19:02 TM.T01 19 tachg_Idle
15:19:35 TM.T01 17 tachg_Delivering
15:20:09 TM.T01 17 tachg_Connected
15:20:16 TM.T01 17 tachg_Clearing
15:20:16 TM.T01 17 IsdnTrunkManager::CallRelease - release Channel.
15:20:16 TM.T01 17 tachg_Idle
15:20:29 TM.T01 16 tachg_Delivering
15:20:42 TM.T01 23 Incoming Call Accepted
15:20:42 TM.T01 23 tachg_TAInboundCall do nothing
15:20:43 TM.T01 23 tachg_Alerting
15:20:45 TM.T01 16 tachg_Connected
15:20:46 TM.T01 23 tachg_Connected