Setting Up Cisco Unified IP Phones using SIP

Session Initiation Protocol (SIP) registrar functionality in Cisco IOS software is an essential part of Cisco Unified SIP Survivable Remote Site Telephony (SRST). According to RFC 3261, a SIP registrar is a server that accepts Register requests and is typically collocated with a proxy or redirect server. A SIP registrar may also offer location services.

Prerequisites for Configuring the SIP Registrar

Complete the prerequisites documented in the "Prerequisites for Configuring Cisco Unified SIP SRST" section on page 9 section in "Cisco Unified SRST Feature Overview" section on page 1.

Restrictions for Configuring the SIP Registrar

See the restrictions documented in the "Restrictions for Configuring Cisco Unified SIP SRST" section on page 10 section in "Cisco Unified SRST Feature Overview" section on page 1.

Information About Configuring the SIP Registrar

Cisco Unified SIP SRST provides backup to an external SIP call control (IP-PBX) by providing basic registrar and call handling services. These services are used by a SIP IP phone in the event of a WAN connection outage when the SIP phone is unable to communicate with its primary SIP proxy. The Cisco Unified SIP SRST device also provides PSTN gateway access for placing and receiving PSTN calls.

Cisco Unified SIP SRST works for the following types of calls:

- Local SIP IP phone to local SIP phone, if the main proxy is unavailable.
- Additional services like class of restriction (COR) for local SIP IP phones to the outgoing PSTN. For example, to block outgoing 1-900 numbers.

How to Configure the SIP Registrar

Configuring the SIP Registrar

The local SIP gateway that becomes the SIP registrar acts as a backup SIP proxy and accepts SIP Register messages from SIP phones. It becomes a location database of local SIP IP phones.

A registrar accepts SIP Register requests and dynamically builds VoIP dial peers, allowing the Cisco IOS voice gateway software to route calls to SIP phones.

If a SIP Register request has a Contact header that includes a DNS address, the Contact header is resolved before the contact is added to the SIP registrar database. This is done because during a WAN failure (and the resulting Cisco Unified SIP SRST functionality), DNS servers may not be available.

SIP registrar functionality is enabled with the following configuration. By default, Cisco Unified SIP SRST is not enabled and cannot accept SIP Register messages. The following configuration must be set up to accept incoming SIP Register messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **allow-connections sip to sip**
5. **sip**
6. **registrar server** [**expires** [**max***sec*] [**min***sec*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service configuration mode.
Step 4	**allow-connections sip to sip** **Example:** `Router(config-voi-srv)# allow-connections sip to sip`	Allows connections from SIP to SIP endpoints.

	Command or Action	Purpose
Step 5	**sip** **Example:** `Router(config-voi-srv)# sip`	Enters SIP configuration mode.
Step 6	**registrar server** [**expires** [**max***sec*] [**min***sec*]] **Example:** `Router(conf-serv-sip)# registrar server expires max 600 min 60`	Enables SIP registrar functionality. The keywords and arguments are defined as follows: • expires: (Optional) Sets the active time for an incoming registration. • max sec: (Optional) Maximum expiration time for a registration, in seconds. The range is from 600 to 86400. The default is 3600. **Note** Ensure that the registration expiration timeout is set to a value smaller than the TCP connection aging timeout to avoid disconnection from the TCP. • min sec: (Optional) Minimum expiration time for a registration, in seconds. The range is from 60 to 3600. The default is 60.
Step 7	**end** **Example:** `Router(conf-serv-sip)# end`	Returns to privileged EXEC mode.

What to do next

For incoming SIP Register messages to be successfully accepted, users must also set up a voice register pool. See the

Configuring Backup Registrar Service to SIP Phones

Backup registrar service to SIP IP phones can be provided by configuring a voice register pool on SIP gateways. The voice register pool configuration provides registration permission control and can also be used to configure some dial-peer attributes that are applied to the dynamically created VoIP dial peers when SIP phone registrations match the pool. The following call types are supported:

SIP IP phone to or from:

- Local PSTN
- Local analog FXS phones
- Local SIP IP phone

The commands in the configuration below provide registration permission control and set up a basic voice register pool. The pool gives users control over which registrations are accepted by a Cisco Unified SIP SRST device and which can be rejected. Registrations that match this pool create VoIP SIP dial peers with the dial-peer attributes set to these configurations. Although only the id command is mandatory, this configuration example shows basic functionality.

For command-level information, see the appropriate command page in Cisco Unified SRST and Cisco Unified SIP SRST Command Reference (All Versions).

Before you begin

The SIP registrar must be configured before a voice register pool is set up. See the

Restrictions

- The **id** command identifies the individual SIP IP phone or sets of SIP IP phones that are to be configured. Thus, the**id**command configured in Step 5 is required and must be configured before any other voice register pool commands. When the**mac***address* keyword and argument are used, the IP phone must be in the same subnet as that of the router's LAN interface, such that the phone's MAC address is visible in the router's Address Resolution Protocol (ARP) cache. Once a MAC address is configured for a specific voice register pool, remove the existing MAC address before changing to a new MAC address.
- Proxy dial peers are autogenerated dial peers that route all calls from the PSTN to Cisco Unified SIP SRST. When a SIP phone registers to Cisco Unified SIP SRST and the **proxy** command is enabled, two dial peers are automatically created. The first dial peer routes to the proxy, and the second (or fallback) dial peer routes to the SIP phone. The same functionality can also be achieved with the appropriate creation of static dial peers (manually creating dial peers that point to the proxy). Proxy dial peers can be monitored to one proxy IP address, only. That is, only one proxy from a voice registration pool can be monitored at a time. If more than one proxy address needs to be monitored, you must manually create and configure additional dial peers.
- If Jabber for desktop clients must register with Unified SRST, ensure that**voice register pools** are configured for all desktop computer networks.

Note To monitor SIP proxies, the **call fallback active** command must be configured, as described in Step 3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call fallback active**
4. **voice register pool** *tag*
5. **id** { **network** *address* **mask** *mask* | **ip** *address* **mask** *mask* | **mac** *address* }
6. **preference** *preference-order*
7. **proxy** *ip-address* [**preference** *value* [**monitor probe {icmp-ping | rtr }** *alternate-ip-address*]]
8. **voice-class codec** *tag*
9. (Optional) **application** *application-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call fallback active** **Example:** `Router(config)# call fallback active`	Enables a call request to fall back to alternate dial peers in case of network congestion. This command is used if you want to monitor the proxy dial peer and fallback to the next preferred dial peer. For full information on the call fallback active command, see PSTN Fallback Feature.
Step 4	**voice register pool** *tag* **Example:** `Router(config)# voice register pool 12`	Enters voice register pool configuration mode for SIP phones. Use this command to control which registrations are accepted or rejected by a Cisco Unified SIP SRST device.
Step 5	**id** { **network** *address* **mask** *mask* | **ip** *address* **mask** *mask* | **mac** *address* } **Example:** `Router(config-register-pool)# id network 172.16.0.0 mask 255.255.0.0`	Explicitly identifies a locally available individual or set of SIP IP phones. The keywords and arguments are defined as follows: • **network** *address* **mask** *mask* : The **network** *address* **mask** *mask* keyword/argument combination is used to accept SIP Register messages for the indicated phone numbers from any IP phone within the indicated IP subnet. • **ip** *address* **mask** *mask* : The **ip** *address* **mask** *mask* keyword/argument combination is used to identify an individual phone. • **mac** *address* : MAC address of a particular Cisco Unified IP Phone.
Step 6	**preference** *preference-order* **Example:** `Router(config-register-pool)# preference 2`	Sets the preference order for the VoIP dial peers to be created. Range is from 0 to 10. Default is 0, which is the highest preference. The preference must be greater (lower priority) than the preference configured with the **preference** keyword in the **proxy** command.
Step 7	**proxy** *ip-address* [**preference** *value* [**monitor probe** **{icmp-ping | rtr }** *alternate-ip-address*]]	Autogenerates additional VoIP dial peers to reach the main SIP proxy whenever a Cisco Unified SIP IP Phone registers

	Command or Action	Purpose
	Example: `Router(config-register-pool)# proxy 10.2.161.187 preference 1`	with a Cisco Unified SIP SRST gateway. The keywords and arguments are defined as follows: • *ip-address* : The *ip-address* of the SIP Proxy. • **preference** *value* : Defines the preference of the proxy dial peers that are created. The preference must be less (higher priority) than the preference configured with the **reference** command. Range is from 0 to 10. The highest preference is 0. There is no default. • **monitor probe** : Enables monitoring of proxy dial peers. • **icmp-ping** : Enables monitoring of proxy dial peers using ICMP ping. **Note** The dial peer on which the probe is configured will be excluded from call routing only for outbound calls. Inbound calls can arrive through this dial peer. • **rtr** : Enables monitoring of proxy dial peers using RTR probes. • *alternate-ip-address* : Enables monitoring of alternate IP addresses other than the proxy address. For example, to monitor a gateway front end to a SIP proxy.
Step 8	**voice-class codec** *tag* **Example:** `Router(config-register-pool)# voice-class codec 15`	Sets the voice class codec parameters. The tag argument is a codec group number between 1 and 10000.
Step 9	(Optional) **application** *application-name* **Example:** `Router(config-register-pool)# application SIP.App`	Selects the session-level application on the VoIP dial peer. Use the *application-name* argument to define a specific interactive voice response (IVR) application.
Step 10	**end** **Example:** `Router(config-register-pool)# end`	Returns to privileged EXEC mode.

What to do next

There are several more voice register pool commands that add functionality, but that are not required. See the for these commands.

Configuring Backup Registrar Service to SIP Phone (Using Optional Commands)

The prior configurations set up a basic voice register pool. The configuration in this procedure adds optional attributes to increase functionality.

Before you begin

- Prerequisites as described in the .
- Configuration of the required commands as described in the .
- Before configuring the **alias** command, translation rules must be set using the **translate-outgoing (voice register pool)** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice register pool** *tag*
4. **translation-profile outgoing** *profile-tag*
5. **alias** *tag pattern* **to** *target* [**preference** *value*]
6. **cor {incoming | outgoing}** *cor-list-name {cor-list-number starting-number [- ending-number] |* **default }**
7. **incoming called-number** *[number]*
8. **number** *tag number-pattern {* **preference** *value }* [**huntstop**]
9. **dtmf-relay [cisco-rtp] [rtp-nte] [sip-notify]**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice register pool** *tag* **Example:** `Router(config)# voice register pool 12`	Enters voice register pool configuration mode. Use this command to control which registrations are accepted or rejected by a Cisco Unified SIP SRST device.

Step 4	**translation-profile outgoing** *profile-tag*
	Example:
	`Router(config-register-pool)#`
	`voice translation-rule 1`
	`rule 1 /1000/ /1006/`
	`!`
		Use this command to apply the translation profile to a specific directory number or to all directory numbers on a SIP phone.
		Profile-tag: Translation profile name to handle translation to outgoing calls.

	Command or Action	Purpose

	`!`
	`voice translation-profile 1`
	`translate called 1`
	`!`
	`voice register pool xxx`
	`translation-profile outgoing 1`	

Step 5	**alias** *tag pattern* **to** *target* [**preference** *value*] **Example:** `Router(config-register-pool)# alias 1 94... to 91011 preference 8`	Allows Cisco Unified SIP IP Phones to handle inbound PSTN calls to telephone numbers that are unavailable when the main proxy is not available. The keywords and arguments are defined as follows: • tag : Number from 1 to 5 and the distinguishing factor when there are multiple alias commands. • *pattern* : The prefix number; matches the incoming telephone number and may include wildcards. • **to**: Connects the tag number pattern to the alternate number. • *target* : The target number; an alternate telephone number to route incoming calls to match the number pattern. • **preference***value* : Assigns a dial-peer preference value to the alias. The *value* argument is the value of the associated dial peer, and the range is from 1 to 10. There is no default.
Step 6	**cor {incoming | outgoing}** *cor-list-name {cor-list-number starting-number [- ending-number] |* **default }** **Example:** `Router(config-register-pool)# cor incoming call91 1 91011`	Configures a class of restriction (COR) on the VoIP dial peers associated with directory numbers. COR specifies which incoming dial peers can use which outgoing dial peers to make a call. Each dial peer can be provisioned with an incoming and outgoing COR list. The keywords and arguments are defined as follows: • **incoming** : COR list to be used by incoming dial peers. • **outgoing** : COR list to be used by outgoing dial peers. • *cor-list-name* : COR list name. • *cor-list-number* : COR list identifier. The maximum number of COR lists that can be created is four, comprised of incoming or outgoing dial peers. • *starting-number* : Start of a directory number range, if an ending number is included. Can also be a standalone number. • Indicator that a full range is configured. • *ending-number* : End of a directory number range.

	Command or Action	Purpose
		• **default**: Instructs the router to use an existing default COR list.
Step 7	**incoming called-number** *[number]* **Example:** `Router(config-register-pool)# incoming called-number 308`	Applies incoming called parameters to dynamically created dial peers. The number argument is optional and indicates a sequence of digits that represent a phone number prefix.
Step 8	**number** *tag number-pattern {* **preference** *value }* [**huntstop**] **Example:** `Router(config-register-pool)# number 1 50.. preference 2`	Indicates the E.164 phone numbers that the registrar permits to handle the Register message from the Cisco Unified SIP IP Phone. The keywords and arguments are defined as follows: • *tag* : Number from 1 to 10 and the distinguishing factor when there are multiple number commands. • *number-pattern* : Phone numbers (including wildcards and patterns) that are permitted by the registrar to handle the Register message from the SIP IP phone. • **preference value** : Defines the number list preference order. • **huntstop** : Stops hunting if the dial peer is busy.
Step 9	**dtmf-relay [cisco-rtp] [rtp-nte] [sip-notify]** **Example:** `Router(config-register-pool)# dtmf-relay rtp-nte`	Specifies how a SIP gateway relays dual tone multifrequency (DTMF) tones between telephony interfaces and an IP network. The keywords are defined as follows: • **cisco-rtp**: Forwards DTMF tones by using Real-Time Transport Protocol (RTP) with a Cisco proprietary payload type. • **rtp-nte**: Forwards DTMF tones by using RTP with the Named Telephone Event (NTE) payload type. • **sip-notify**: Forwards DTMF tones using SIP NOTIFY messages.
Step 10	**end** **Example:** `Router(config-register-pool)# end`	Returns to privileged EXEC mode.

Example

The following partial output from the show running-config command shows that voice register pool 12 is configured to accept all registrations from SIP IP phones with extension number 50xx from the 172.16.0.0/16 network. Autogenerated dial peers for registrations that match pool 12 have attributes configured in this pool.

```
.
.
.
voice register pool 12
id network 172.16.0.0 mask 255.255.0.0
number 1 50.. preference 2
application SIP.app
preference 2
incoming called-number
cor incoming allowall default
translate-outgoing called 1
voice-class codec 1
.
.
.
```

Verifying SIP Registrar Configuration

To help you troubleshoot a SIP registrar and voice register pool, perform the following steps.

SUMMARY STEPS

1. **debug voice register errors**
2. **debug voice register events**
3. **show sip-ua status registrar**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**debug voice register errors**
	Example:
	`Router# debug voice register errors`
	`*Apr 22 11:52:54.523 PDT: VOICE_REG_POOL: Contact doesn't match any pools`
	`*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Register request for (33015) from (10.2.152.39)`
	`*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Contact doesn't match any pools.`
	`*Apr 22 11:52:54.559 PDT: VOICE_REG_POOL: Register request for (33017) from (10.2.152.39)`
	`*Apr 22 11:53:04.559 PDT: VOICE_REG_POOL: Maximum registration threshold for pool(3) hit`
		Use this command to debug errors that happen during registration.
		If there are no voice register pools configured for a particular registration request, the message `Contact doesn’t match any pools` is displayed.

Step 2	**debug voice register events**
	Example:
	`Router# debug voice register events`
	`Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Contact matches pool 1`
	`Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: key(91011) contact(192.168.0.2) add to contact table`
	`Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: key(91011) exists in contact table`
	`Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: contact(192.168.0.2) exists in contact table, ref updated`
		Using the **debug voice register events** command should suffice to display registration activity. Registration activity includes matching of pools, registration creation, and automatic creation of dial peers. For more details and error conditions, you can use the **debug voice register errors** command.
		The phone number 91011 registered successfully, and type 1 is reported, which means there is a pre-existing VoIP dial peer.

	Command or Action	Purpose
	`Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Created dial-peer entry of type 1` `Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Registration successful for 91011, registration id is 257`	

Step 3	**show sip-ua status registrar**
	Example:
	`Router# show sip-ua status registrar`
	`Line    destination expires(sec) contact`
	`======= =========== ============ =======`
	`91021   192.168.0.3 227          192.168.0.3`
	`91011   192.168.0.2 176          192.168.0.2`
	`95021   10.2.161.50 419          10.2.161.50`
	`95012   10.2.161.50 419          10.2.161.50`
	`95011   10.2.161.50 420          10.2.161.50`
	`95500   10.2.161.50 420          10.2.161.50`
	`94011   10.2.161.40 128          10.2.161.40`
	`94500   10.2.161.40 129          10.2.161.40`
		Use this command to display all the SIP endpoints currently registered with the contact address.

Verifying Proxy Dial-Peer Configuration

To use the **icmp-ping** keyword with the **proxy** command to assist in troubleshooting proxy dial peers, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **voice register pool**
3. **proxy** *ip-address*[**preference***value*] [**monitor probe {icmp-ping|rtr}**[*alternate-ip-address*]]
4. **end**
5. **show voice register dial-peers**
6. **show dial-peer voice**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Use this command to enter global configuration mode.
Step 2	**voice register pool** **Example:** `Router(config)# voice register pool 1`	Use this command to enter voice register pool configuration mode.
Step 3	**proxy** *ip-address*[**preference***value*] [**monitor probe {icmp-ping|rtr}**[*alternate-ip-address*]] **Example:**	Set the **proxy** command to monitor with **icmp-ping**.

	Command or Action	Purpose
	`Router(config-register-pool)# proxy 10.2.161.187 preference 1 monitor probe icmp-ping`	
Step 4	**end** **Example:** `Router(config-register-pool)# end`	Returns to privileged EXEC mode.

Step 5	**show voice register dial-peers**
	Example:
	`Router# show voice register dial-peers`
	`dial-peer voice 40035 voip`
	`preference 5`
	`destination-pattern 91011`
	`session target ipv4:192.168.0.2`
	`session protocol sipv2`
	`voice-class codec 1`
	`dial-peer voice 40036 voip`
	`preference 1`
	`destination-pattern 91011`
	`session target ipv4:10.2.161.187`
	`session protocol sipv2`
	`voice-class codec 1`
	`monitor probe icmp-ping 10.2.161.187`
		Use this command to verify dial-peer configurations, and notice that **icmp-ping** monitoring is set.

Step 6	**show dial-peer voice** **Example:** `Router# show dial-peer voice` `VoiceOverIpPeer40036` `peer type = voice, information type = voice,` `description = ``',` `tag = 40036, destination-pattern = ``91011',` `answer-address = ``', preference=1,` `CLID Restriction = None` `CLID Network Number = ``'` `CLID Second Number sent` `source carrier-id = ``', target carrier-id = ``',` `source trunk-group-label = ``', target` `trunk-group-label = ``',` `numbering Type = ``unknown'` `group = 40036, Admin state is up, Operation state` ` is` `up,` `incoming called-number = ``', connections/maximum` `=` `0/unlimited,` `! Default output for incoming called-number command` `DTMF Relay = disabled,` `modem transport = system,` `huntstop = disabled,` `in bound application associated: 'DEFAULT'` `out bound application associated: ''` `dnis-map =` `permission :both` `incoming COR list:maximum capability` `! Default output for cor command` `outgoing COR list:minimum requirement` `! Default output for cor command` `Translation profile (Incoming):`	Use the show dial-peer voice command on dial peer 40036, and notice the monitor probe status. **Note** Also highlighted is the output of the **cor** and **incoming called-number** commands.

	Command or Action	Purpose
	Translation profile (Outgoing): incoming call blocking: translation-profile = `' disconnect-cause = `no-service' advertise 0x40 capacity_update_timer 25 addrFamily 4 oldAddrFamily 4 type = voip, session-target = `ipv4:10.2.161.187', technology prefix: settle-call = disabled ip media DSCP = ef, ip signaling DSCP = af31, ip video rsvp-none DSCP = af41,ip video rsvp-pass DSCP = af41 ip video rsvp-fail DSCP = af41, UDP checksum = disabled, session-protocol = sipv2, session-transport = system, req-qos = best-effort, acc-qos = best-effort, req-qos video = best-effort, acc-qos video = best-effort, req-qos audio def bandwidth = 64, req-qos audio max bandwidth = 0, req-qos video def bandwidth = 384, req-qos video max bandwidth = 0, RTP dynamic payload type values: NTE = 101 Cisco: NSE=100, fax=96, fax-ack=97, dtmf=121, fax-relay=122 S=123, ClearChan=125, PCM switch over u-law=0,A-law=8 RTP comfort noise payload type = 19 fax rate = voice, payload size = 20 bytes fax protocol = system fax-relay ecm enable fax NSF = 0xAD0051 (default) codec = g729r8, payload size = 20 bytes, Media Setting = flow-through (global) Expect factor = 0, Icpif = 20, Playout Mode is set to adaptive, Initial 60 ms, Max 300 ms Playout-delay Minimum mode is set to default, value 40 ms Fax nominal 300 ms Max Redirects = 1, signaling-type = cas, VAD = enabled, Poor QOV Trap = disabled, Source Interface = NONE voice class sip url = system, voice class sip rel1xx = system, monitor probe method: icmp-ping ip address: 10.2.161.187, Monitored destination reachable voice class perm tag = `' Time elapsed since last clearing of voice call statistics never Connect Time = 0, Charged Units = 0, Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0 Accepted Calls = 0, Refused Calls = 0, Last Disconnect Cause is "", Last Disconnect Text is "", Last Setup Time = 0.	

What to do next

The next step is configuring incoming and outgoing calls for Cisco Unified SRST. For more information,