



## Cisco Unified Border Element (SP Edition) Registration Features

---

Cisco Unified Border Element (SP Edition) supports the SIP Fast Registration, SoftSwitch Shielding, Registration Monitoring, Aggregate Registration, Provisioned Delegate Registration, and Contact Username Passthrough features in the unified model.

SIP Fast Registration addresses the problem where SIP messages to the Network Address Translation (NAT) endpoint are unable to penetrate the NAT and firewalls to establish calls. Using SIP Fast Registration, the NAT endpoints transmit SIP REGISTER requests at a high enough frequency to keep the NAT pinhole alive.

The SoftSwitch Shielding feature allows a lower SIP registration rate on the links to registrars (typically softswitches) than on the links to endpoints. Allowing a lower registration rate shields the softswitch from an undesirably high rate of re-registrations.

Cisco Unified Border Element (SP Edition) supports monitoring events subscription for changes of the registration state with the Registration Monitoring functionality.

Aggregate Registration registers all the endpoints connected to it in a single registration. This functionality enables Cisco Unified Border Element (SP Edition) to support devices that implicitly register multiple endpoints through it.

The Provisioned Delegate Registration feature allows the Cisco Unified Border Element (SP Edition) to support client or end user devices that cannot register themselves in a network where SIP calls are passing through a registrar. Cisco Unified Border Element (SP Edition) is able to register on behalf of such client devices. The Provisioned Delegate Registration feature can support Cisco Telepresence systems where the end user applications cannot send the registration message and Cisco Unified Border Element (SP Edition) does it on their behalf.

The Contact Username Passthrough enhancement enables interoperability with softswitches that require the contact username portion of the Contact URI in SIP REGISTER requests to pass through unchanged.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

[http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu\\_book.html](http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html).

For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or a Cisco IOS master commands list.



### Note

---

These features are supported in the unified model.

---

**Feature History for Registration Features**

| <b>Release</b>            | <b>Modification</b>  |
|---------------------------|--|
| Cisco IOS XE Release 2.4  | The SIP Fast Registration, SoftSwitch Shielding, Registration Monitoring, Aggregate Registration, and Delegate Registration features were introduced on the Cisco IOS XR along with support for the unified model. |
| Cisco IOS XE Release 2.5  | The Contact Username Passthrough for non-IMS networks and Support for Supported Path Under REGISTER Request features were added.   |
| Cisco IOS XE Release 3.1S | The Per Subscriber Delete feature was added.   |
| Cisco IOS XE Release 3.2S | The adding expires-header to register-message feature was added.   |
| Cisco IOS XE Release 3.3S | The Alternative Contact Rewriting feature was added.   |

# Contents

This chapter contains the following sections:

- [Prerequisites, page 22-3](#)
- [Restrictions, page 22-3](#)
- [Information About SIP Registration, page 22-3](#)
- [Adding an Expires-Header to a Register-Message, page 22-4](#)
- [Support for Supported Path Under REGISTER Request, page 22-6](#)
- [Information About Contact Username Passthrough, page 22-6](#)
- [Configuring Contact Username Passthrough, page 22-7](#)
- [Information About Alternative Contact Rewriting, page 22-9](#)
- [Configuring Alternative Contact Rewriting, page 22-10](#)
- [Information About SIP Fast Registration, page 22-12](#)
- [Configuring SIP Fast Registration, page 22-15](#)
- [Information About SoftSwitch Shielding, page 22-16](#)
- [Configuring SoftSwitch Shielding, page 22-17](#)
- [Information About Registration Monitoring, page 22-19](#)
- [Configuring Registration Monitoring, page 22-20](#)
- [Information About Per Subscriber Delete, page 22-21](#)
- [Configuring Per Subscriber Delete, page 22-21](#)
- [Information About Aggregate Registration, page 22-22](#)
- [Configuring Aggregate Registration, page 22-22](#)
- [Information About Provisioned Delegate Registration, page 22-24](#)
- [Provisional Delegate Registration Commands, page 22-26](#)
- [Configuration Examples, page 22-30](#)

## Prerequisites

The following prerequisite is required to implement SoftSwitch Shielding, Registration Monitoring, Aggregate Registration, and Provisioned Delegate Registration:

Before implementing these features, Cisco Unified Border Element (SP Edition) must already be configured.

## Restrictions

SIP Fast Registration has the following restrictions:

- Only UDP is supported.
- REGISTERs with a zero expiry time (“Unregisters”) are always forwarded to the registrar and not fast-pathed, if the SBC matches them to a known registration.
- Minimal parsing of REGISTER requests is performed before a decision is taken to send a fast-path response; this minimizes the load on the SBC. A REGISTER request is only fast-pathed if its expiry interval is not zero and it comes from the same IP address and port as a known subscription.
- Endpoints that send their requests from ephemeral (short-lived) ports do not have their registration requests fast-pathed.
- The “FastReg interval” cannot be higher than the “MinExpiry interval.” If the “MinExpiry interval” is less than twice the “FastReg interval,” fast-pathing is not performed.

Provisioned Delegate Registration has the following restrictions:

- The delegate registration configuration is limited to no more than 1000 subscribers with each subscriber having no more than 5 contacts.
- H.323 adjacencies and SIP to H.323 interworking are not supported in Cisco IOS XE Release 2.4 and earlier.

## Information About SIP Registration

Registration is required if the end user has a dynamic IP address, if the provider does not support static hostnames, or if NAT is used.

In a SIP REGISTER message, the Contact: header contains the URI that identifies the subscriber.

When a device registers in a non-IMS network, Cisco Unified Border Element (SP Edition) takes the SIP REGISTER Contact: header and modifies it by replacing the contact username with a hash to produce a unique contact username. This is the default behavior for a typical registration. This is needed because there may be multiple UNI adjacencies in different VLANs, which have similar contacts. (Note that this does not apply to the IMS P-CSCF profile.) Then, the SBC forwards the REGISTER to the registrar containing this new modified Contact: header. Meanwhile, the SBC will also store a record of the original contact and the modified contact in its internal memory.

When the core network desires to ring that subscriber, the SBC will receive an INVITE containing the modified contact information. The Cisco SBC will check its memory to look up the information, and will swap out the header with the original information, and will direct the call to the appropriate SIP adjacency in the correct customer network.

This means that no explicit call routing detail needs to be configured in the call policy for routing calls from the core to subscribers, since the SBC has its internal memory of registrations.

Note that even though a SIP adjacency may be intended to receive only subscriber (registered) traffic, it is still possible for unregistered callers to initiate calls from that same adjacency. This can be considered useful, because emergency callers therefore may not need to register first.

When the call arrives at the softswitch, it can check if the subscriber has registered or not, and if to allow the call or not.

In the case where softswitch interoperability is desired, you may want to pass through the contact username instead of hashing it. Cisco Unified Border Element (SP Edition) provides a Contact Username Passthrough enhancement for non-IMS networks. See [Information About Contact Username Passthrough](#), page 22-6.

## Adding an Expires-Header to a Register-Message

Some registrars or endpoints might fail to understand the Expires parameters configured in the contact URI. To overcome this issue, you can configure the SBC to add an Expires header to the register messages.

In SIP, the expiry time of a registration is specified by registering the endpoints using the following:

- Expires header
- The Expires parameter on the registered contact URI.
- The registrar can choose an expiration period, if no expirations period is specified.

## Configuring SBC to Add an Expires-Header

To configure SBC to add an Expires header to the register messages, complete the following steps:

### SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **expires-header**
6. **softswitch-shield**
7. **exit**
8. **end**
9. **show sbc** *sbc-name* **sbe** **adjacencies** *adjacency-name* **Detail**

## DETAILED STEPS

|        | Command or Action   | Purpose   |
|--------|---|---|
| Step 1 | <b>configure</b><br><br><b>Example:</b><br>Router# configure  | Enables the global configuration mode.  |
| Step 2 | <b>sbc service-name</b><br><br><b>Example:</b><br>Router(config)# sbc mysbc   | Enters the mode of an SBC service.<br><br>Use the <i>service-name</i> argument to define the name of the service.   |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe  | Enters the mode of an SBE entity within an SBC service.   |
| Step 4 | <b>adjacency sip adjacency-name</b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip sipGW                 | Enters the mode of an SBE SIP adjacency.<br><br>Use the <i>adjacency-name</i> argument to define the name of the service.   |
| Step 5 | <b>expires-header options</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# expires-header<br>add-not-present | Adds the Expires parameter in a SIP contact header.<br><br>Use the <i>options</i> argument to specify one of the following strings for adding expires to the header: <ul style="list-style-type: none"> <li>• <i>add-not-present</i>—The SBC provides expiry information in the format provided by an endpoint, or as indicated by other configurations.</li> <li>• <i>add-smallest</i>—The value of the Expires header is set to the value of the smallest Expires parameter in any provided contact.</li> <li>• <i>add-value</i>—The SBC adds an Expires header to any REGISTER request that is sent out on the specified adjacency that does not contain an expiry value.</li> </ul> |
| Step 6 | <b>softswitch-shield</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)#<br>softswitch-shield                   | Enables softswitch shielding on the SIP.  |
| Step 7 | <b>exit</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip-ping)# exit   | Exits the adj-sip-ping mode, and moves to adj-sip mode.   |

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 8 | <b>end</b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# end   | Exits the SBE mode and returns to the privileged EXEC mode.       |
| Step 9 | <b>show sbc sbc-name sbe adjacencies adjacency-name detail</b><br><br><b>Example:</b><br>Router# show sbc mysbc sbe adjacencies sipGW detail | Lists the configured Expires headers for the specified adjacency. |

## Support for Supported Path Under REGISTER Request

Starting with Cisco IOS XE Release 2.5, Cisco Unified Border Element (SP Edition) supports the use of the Path extension header field in the Supported field of a REGISTER Request. The Path field provides a way to accumulate and send a list of proxies between a SIP user agent and a registrar. For information on the Path field, see RFC 3327 *Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts*.

## Information About Contact Username Passthrough

The Contact Username Passthrough feature enables interoperability with softswitches that require the contact username portion of the Contact URI in SIP REGISTER requests to pass through unchanged. In certain situations in non-IMS networks, a softswitch may be unable to operate with Cisco Unified Border Element (SP Edition) rewriting or hashing the contact username portion of the Contact URI in SIP REGISTER requests. In these cases, subscribers may not be able to register through the SBC unless you configure the SBC to pass through the contact username.

In a typical SIP registration process, the default behavior is that Cisco Unified Border Element (SP Edition) rewrites the URIs in the Contact headers of REGISTER requests sent by subscribers for the following reasons:

- To remain on the signaling path for requests sent to this subscriber.
- To disambiguate subscribers that register from different devices with the same private username by replacing the username part of the Contact URI with a unique string.

For example, the SBC receives two REGISTER requests, with the following contact URIs using the same username, “bob”:

```
bob@1.1.1.1
bob@2.2.2.2
```

In each case, the REGISTER contains a Contact URI with the SBC’s address. The SBC replaces or rewrites the username “bob” in each URI with a unique string to disambiguate them.

In Cisco IOS XE Release 2.5 and later, you can choose to configure the SBC to pass through, not rewrite, the contact username on SIP REGISTER requests by ensuring that each contact username associated with a given subscriber uses a different port number. By using unique ports for each contact sent to the registrar, the SBC can uniquely correlate to the registered endpoints without requiring a unique username. This can be configured for each adjacency facing the registrar.

The following is an example of contact username passthrough where the username “bob” is passed through unchanged and the hostport is rewritten with the address of the SBC and a unique port number:

```
sip:bob@1.1.1.1 -----> sip:bob@192.168.101.1:5060
```

See the [?\\$paranum>Contact Username Passthrough Examples?](#) section on page 22-38 for more configuration examples.

**Note**

This feature has no effect in IMS deployments where the SBC does not rewrite contact usernames.

## Configuring Contact Username Passthrough

You can use the **registration contact username passthrough** and the **signaling-port** commands in the (config-sbc-sbe-adj-sip) configuration mode to configure the Contact Username Passthrough feature.

The **registration contact username** command with the **passthrough** option allows you to specify that the contact username in the SIP REGISTER request should be passed through unchanged when rewriting contacts. This option should be enabled on the registrar-facing adjacency. The **passthrough** option disambiguates subscribers that register from different devices with the same private username by using a unique local port number when multiple contact URIs are registered for the same public ID. The range of valid signaling ports are configured with the **signaling-port** command on a registrar-facing adjacency.

If you do not specify the *max-port-num* option in the **signaling-port** command on this adjacency, then the SBC is not able to disambiguate subscribers that register from different devices with the same username.

The default is the **rewrite** option which allows the username to be changed when rewriting contacts.

**Note**

If the contact username is longer than 32 characters, then it is *not* passed through and is replaced with a hash as is the case when the default **rewrite** option is chosen.

The following example configures the SBC to specify that the contact username is passed through unchanged.

### SUMMARY STEPS

1. **configure terminal**
2. **sbc *sbc-name***
3. **sbe**
4. **adjacency sip *adjacency-name***
5. **no attach**
6. **registration contact username {passthrough | rewrite}**
7. **signaling-port *port-num* [*max-port-num*]**
8. **exit**
9. **end**
10. **show sbc sbe adjacencies**

## DETAILED STEPS

|        | Command or Action   | Purpose   |
|--------|---|---|
| Step 1 | <code>configure terminal</code><br><br><b>Example:</b><br>Router# <code>configure terminal</code>   | Enables global configuration mode.  |
| Step 2 | <code>sbc sbc-name</code><br><br><b>Example:</b><br>Router(config)# <code>sbc mySbc</code>  | Creates the SBC service on the SBC and enters into SBC configuration mode.  |
| Step 3 | <code>sbe</code><br><br><b>Example:</b><br>Router(config-sbc)# <code>sbe</code>   | Enters the mode of the signaling border element (SBE) function of the SBC.  |
| Step 4 | <code>adjacency sip adjacency-name</code><br><br><b>Example:</b><br>Router(config-sbc-sbe)# <code>adjacency sip adj1</code>   | Configures the adjacency (facing the registrar), and enters into adjacency sip configuration mode.  |
| Step 5 | <code>no attach</code><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# <code>no attach</code>   | (Optional) Use this command to detach an existing adjacency so it is not active before modifying it.  |
| Step 6 | <code>registration contact username {passthrough rewrite}</code><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# <code>registration contact username passthrough</code> | Specifies whether the contact username in the SIP REGISTER request is passed through unchanged when rewriting contacts.<br><br>This option must be enabled on the registrar-facing adjacency.<br><br>The <b>passthrough</b> keyword disambiguates subscribers that register from different devices with the same private username by using a unique local port number when multiple contact URIs are registered for the same public ID.   |
| Step 7 | <code>signaling-port port-num [max-port-num]</code><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# <code>signaling-port 5060 5062</code>                               | Configures range of valid signaling ports on a registrar-facing adjacency to allow the SBC to disambiguate subscribers that register from different devices with the same username.<br><br><i>max-port-num</i> is the range from 1 through 65535.<br><br>If both <i>port-num</i> and <i>max-port-num</i> are specified, then the <i>port-num</i> indicates the lower boundary of the range and <i>max-port-num</i> indicates the upper boundary of the range. If no <i>max-port-num</i> is specified, then the adjacency listens only on the single <i>port-num</i> . <i>Max-port-num</i> only needs to be set if a range of local listen ports is required for this adjacency. |

|         | Command or Action  | Purpose  |
|---------|--|--|
| Step 8  | <code>exit</code><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# exit                 | Exits adjacency sip configuration mode and enters into SBE configuration mode. |
| Step 9  | <code>end</code><br><br><b>Example:</b><br>Router(config-sbc-sbe)# end                           | Exits SBE configuration mode and returns to privileged EXEC mode.              |
| Step 10 | <code>show sbc sbe adjacencies</code><br><br><b>Example:</b><br>Router# show sbc sbe adjacencies | Displays SBC adjacencies.  |

## Information About Alternative Contact Rewriting

In a User-to-Network Interface (UNI) deployment scenario, the endpoint registers to registrar, softswitch, or proxy through the SBC. The SBC rewrites the received contact header to use its own signaling address and keeps itself in the call signaling flow. The SBC maintains the mapping between the received and forwarded contact information. This ensures that the contacts received from the multiple devices are unique, and provides anonymity to the subscribers.

Prior to Cisco IOS XE Release 3.3S, the SBC would rewrite the contact by replacing the entire contact with an alphanumeric string generated by hashing the received contact information. However, registrars can determine that the multiple registrations are for the same Address of Record (AOR) by comparing an initial section of the user-info in the contact header. For example, they determine that the two contacts for 02083661177-abc@sbc.com and 02083661177-xyz@sbc.com are for the same endpoint.

From Cisco IOS XE Release 3.3S, the SBC rewrites the contact header in the following two methods:

- Hashed value of hexadecimal characters—`<DN> + "-" + <hashed_value>`, the `<hashed_value>` is a randomly generated value and is unique for a specific endpoint, so that the softswitch can identify those endpoints and initiate forking. Forking is an multiple calls attempt to the endpoints for a single AoR.
- Username of rewritten contact Uniform Resource Identifier (URI) only includes numeric hashed value.

### Delegate Registration

When a delegate subscriber is configured on a preset-access adjacency, the contact header sent to the registrar is rewritten similar to the contact header received on a REGISTER message. Therefore, the Alternative Contact Rewriting feature applies to a delegate registration also. The *original contact* provided to the user on exit and used to generate the rewritten contact is the contact that is configured using the `sip-contact contact uri` command under the SBE Subscriber Entry mode.

### Restrictions on Alternative Contact Rewriting

The feature has the following restriction:

- If there is no username in a contact URI, 32 digits hashed username is used. However, if the original username is 24 bytes or more in length, the username is rewritten in the `<23 digit numeric hash>-<8 digit numeric hash>` format.

## Configuring Alternative Contact Rewriting

This task explains how to configure the Alternative Contact Rewriting feature.

### SUMMARY STEPS

1. **configure**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency** {**sip** | **h323**} *adjacency-name*
5. **registration contact username rewrite** [**numeric** | **userid-and-numeric**]
6. **end**
7. **show sbc** *sbc-name* **sbe** **adjacencies** *adjacency-name* **detail**

### DETAILED STEPS

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 1 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal   | Enables global configuration mode.  |
| Step 2 | <b>sbc</b> <i>sbc-name</i><br><br><b>Example:</b><br>Router(config)# sbc mySbc   | Enters the SBC service mode. Use the <i>sbc-name</i> argument to define the name of the service.  |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe   | Enters the SBE configuration mode.  |
| Step 4 | <b>adjacency</b> { <b>sip</b>   <b>h323</b> } <i>adjacency-name</i><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip pe42 | Configures the adjacency facing the registrar, and enters into adjacency sip configuration mode.<br><br><b>Note</b> The Alternative Contact Support feature does not support the H.323 adjacencies. |

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 5 | <pre>registration contact username rewrite [numeric   userid-and-numeric]]</pre> <p><b>Example:</b><br/>Router(config-sbc-sbe-adj-sip)# registration<br/>contact username rewrite userid-and-numeric</p> | <p>Configures the contact username in a SIP REGISTER request so that it can be modified.</p> <ul style="list-style-type: none"> <li>• <b>rewrite</b>—Allows the contact username in a SIP REGISTER request to be changed or rewritten.</li> <li>• <b>numeric</b>—Rewrites the contact username in a SIP REGISTER request as an originating hashed numeric value.</li> <li>• <b>userid-and-numeric</b>—Rewriting the contact username in a SIP REGISTER request as an originating userid plus hashed numeric value.</li> </ul> |
| Step 6 | <pre>end</pre> <p><b>Example:</b><br/>Router(config-sbc-sbe)# end</p>  | <p>Exits adjacency SIP configuration mode and returns to Exec mode.</p>   |
| Step 7 | <pre>show sbc sbc-name sbe adjacencies adjacency-name detail</pre> <p><b>Example:</b><br/>Router# show sbc sbe mySBC sbe adjacencies pe42</p>  | <p>Displays the detailed field output for the specified SIP adjacency.</p>  |

The following example show the output of the **show sbc sbe adjacencies detail** command:

```
Router# show sbc pe41 sbe adjacencies pe42 detail
SBC Service "pe41"
Adjacency pe42 (SIP)
  Status:                Attached
  Signaling address:     88.41.41.41:5060
  IPsec server port:     0
  Signaling-peer:        88.42.42.42:5060 (Default)
  Signaling-peer status: Not Tested
  Signaling-peer priority: 2147483647
  Signaling-peer switch: always
  Peer status:           Not Tested
  Current peer index:    0
  Force next hop:        No
  Force next hop select: Out-of-dialog
  Admin Domain:         None
  Account:
  Group:                 None
  .
  .
  .
  Rewrite REGISTER:     Off
  Register contact username: Rewrite as userid and digits
  Target address:        None
  NAT Status:            Auto Detect
  Reg-min-expiry:        3000 seconds
  Local Jitter Ratio:    0/1000
  .
  .
  .
```

## Information About SIP Fast Registration

SIP Fast Registration performs the following functions:

- Prompts endpoints to register frequently with Cisco Unified Border Element (SP Edition), causing NAT/Firewall pinholes to remain open.
- Protects internal network elements from the large number of REGISTER messages arising from endpoint registration.
- Allows Cisco Unified Border Element (SP Edition) to do minimal processing of REGISTER messages, which improves performance. This is particularly important when dealing with significant oversubscription—where there typically may be 10 times more subscribers than active calls.

When Cisco Unified Border Element (SP Edition) faces the customer premise side and the customer on the network edge deploys Network Address Translation (NAT) and firewalls, SIP INVITEs to the NAT endpoint are unable to penetrate the NAT to establish calls. To overcome the problem, the endpoints transmit SIP REGISTER requests at a high enough frequency to keep the NAT pinhole alive. The SIP Fast Registration feature off loads the processing from the registrar for a large number of endpoints and generates SIP REGISTER replies from the data plane (forwarding services provided by Cisco QuantumFlow Processor (QFP)). This also limits the impact on the router's CPU load. The QFP processes the expected SIP re-register messages by short-cutting the REGISTER messages and quickly turning them around.

Typically the registrar responds to the first SIP REGISTER message asking the end-point to send its next SIP REGISTER message within 3600 seconds (configurable). With the SIP Fast Registration feature, Cisco Unified Border Element (SP Edition) intercepts this Reply and informs the end-point to REGISTER every 30 seconds to keep the NAT open. The Route Processor (RP) programs a SIP Fast Registration (SFX) entry with a fast expiry times parameter in seconds. When the fast expiry times parameter expires, the QFP punts the SIP REGISTER message to the RP to update the state before forwarding it to the registrar.

Fast registration is configured per SIP adjacency, on the endpoint-facing adjacency, that is, the adjacency which receives the incoming REGISTER request. After an endpoint has registered using fast registration through an adjacency, all subsequent registration requests from the same endpoint are responded to by Cisco Unified Border Element (SP Edition), without notifying the softswitch, until the registration interval has almost expired.

The following shows a fast registration configuration example:

```
...
Reg-min-expiry:      300 seconds
Fast-register:      Enabled
Fast-register-interval: 60 seconds
Register aggregate: Disabled
...
```

In the example above, Cisco Unified Border Element (SP Edition) performs fast registrations at 60 second intervals, and will send registrations towards the registrar/softswitch at intervals of 480 seconds. This is calculated by a hard-coded algorithm of  $(3 \times \text{fast-register-interval}) + (\text{reg-min-expiry})$ , plus taking into account the expiry time in the inbound registration from the endpoint.

In the above example, Softswitch Shielding is not enabled. Thus if incoming expiry time from the endpoint is 400 seconds, which is less than 480 seconds, then the incoming registration interval to the registrar/softswitch is calculated as 480 seconds. However, if the incoming expiry time from the endpoint is 600 seconds, which is larger than 480 seconds, then the incoming registration interval to the registrar/softswitch is calculated as 600 seconds.

On the other hand, when Softswitch Shielding is enabled, then the Softswitch Shielding timer takes precedence and is always used as the incoming registration interval to the registrar/softswitch. See the [Information About SoftSwitch Shielding? section on page 22-16](#).

When fast registration is enabled, the incoming registration time should not be less than the `fast-register-interval`. Otherwise, the SBC will reject the registration with error message 423 (Interval Too Brief). The SBC compares incoming registration time with the interval set in `fast-register-interval`, instead of the interval in `reg-min-expiry`. If fast registration is *disabled*, then the incoming registration time should not be less than the `reg-min-expiry` time. Otherwise, the SBC will reject the registration with response code 423 (Interval Too Brief).

For information on commands, such as `fast-register-interval` and `reg-min-expiry`, see the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model*.

SIP Fast Registration is not enabled by default. You must configure it with the `inherit profile preset-access` command, either as a global configuration or per adjacency.

Once SIP Fast Registration is configured, fast-pathing is on by default on an adjacency. You can then disable Fast Registration using the `fast-register disable` command.

REGISTER messages are rejected by Cisco Unified Border Element (SP Edition) with a 423 Interval Too Brief response code under the following conditions:

- If Fast Registration is enabled and the Expires header in the REGISTER message is less than the inbound adjacency's "`fast-register-interval`."

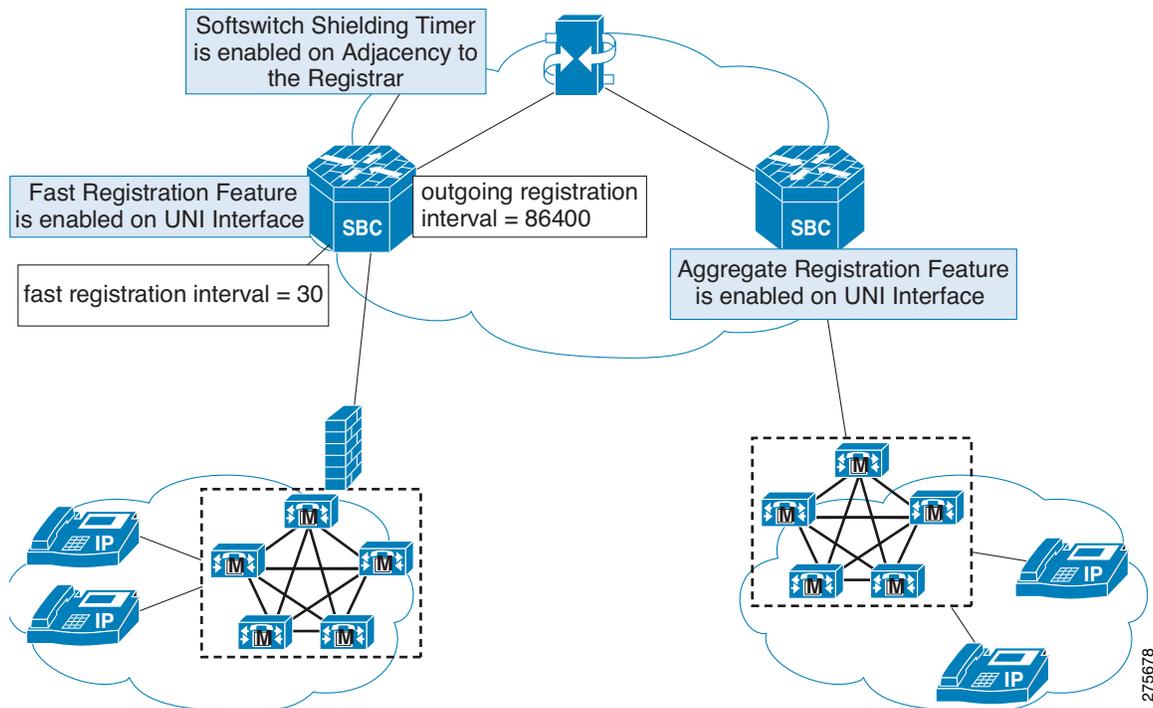
The `fast-register-interval` command controls the recommended rate at which endpoints send REGISTER requests. The lower this value, the more frequently endpoints re-register, thus keeping a NAT/firewall pinhole open. Therefore, we recommend configuring this to a slightly lower value than the pinhole timeout, if that is known.

- If Fast Registration is not enabled and the Expires header in the REGISTER messages is less than the inbound adjacency's "`reg-min-expiry`."

The `reg-min-expiry` command controls the rate at which REGISTER requests are sent from the SBC to the SIP registrar. The lower this value, the greater the potential register load on the softswitch. If fast-pathing is not enabled for an adjacency, SBC rejects any REGISTER requests with a shorter expiry interval than the `reg-min-expiry` command.

Figure 22-1 illustrates where network elements are located in a network configured with Fast Registration, SoftSwitch Shielding, and Aggregate Registration.

**Figure 22-1** Voice Network Elements in a Fast Registration, SoftSwitch Shielding, and Aggregate Registration Network



## Restrictions for SIP Fast Registration

The restrictions for SIP Fast Registration are the following:

- Only UDP is supported.
- REGISTERs with a zero expiry time (“Unregisters”) are always forwarded to the registrar and not fast-pathed, if the SBC matches them to a known registration.
- Minimal parsing of REGISTER requests is performed before a decision is taken to send a fast-path response; this minimizes the load on the SBC. A REGISTER request is only fast-pathed if its expiry interval is not zero and it comes from the same IP address and port as a known subscription.
- Endpoints that send their requests from ephemeral (short-lived) ports do not have their registration requests fast-pathed.
- The fast-register-interval cannot be higher than the reg-min-expiry (minimum expiry value). If the minimum expiry value is less than twice the fast-register-interval, fast-pathing is not performed.

## Configuring SIP Fast Registration

This task configures a basic SIP Fast Registration on an adjacency.

### SUMMARY STEPS

1. **configure**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency** {**sip** | **h323**} *adjacency-name*
5. **inherit profile** {**preset-access** | **preset-core** | **preset-ibcf-ext-untrusted** | **preset-ibcf-external** | **preset-ibcf-internal** | **preset-p-cscf-access** | **preset-p-cscf-core** | **preset-peering** | **preset-standard-non-ims**}
6. **exit**
7. **end**
8. **show platform hardware qfp active feature sbc sfx**

### DETAILED STEPS

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 1 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal   | Enables global configuration mode.   |
| Step 2 | <b>sbc</b> <i>sbc-name</i><br><br><b>Example:</b><br>Router(config)# sbc mySbc   | Creates the SBC service on the SBC and enters into SBC configuration mode.   |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe   | Enters the mode of the signaling border element (SBE) function of the SBC.   |
| Step 4 | <b>adjacency</b> { <b>sip</b>   <b>h323</b> } <i>adjacency-name</i><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip adj1   | Configures the adjacency (facing the subscriber), and enters into adjacency sip configuration mode.<br><br><b>Note</b> H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.  |
| Step 5 | <b>inherit profile</b> { <b>preset-access</b>   <b>preset-core</b>   <b>preset-ibcf-ext-untrusted</b>   <b>preset-ibcf-external</b>   <b>preset-ibcf-internal</b>   <b>preset-p-cscf-access</b>   <b>preset-p-cscf-core</b>   <b>preset-peering</b>   <b>preset-standard-non-ims</b> }<br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# inherit profile preset-access | Configures an inherit profile for the SIP adjacency.<br><br>The SIP adjacency must be configured to preset-access for Fast Register. An access adjacency faces user equipment, such as a subscriber's telephone or other SIP device, that attempts to register through the SBC.<br><br>The default is preset-core. |

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 6 | <pre>exit</pre> <p><b>Example:</b><br/>Router(config-sbc-sbe-adj-sip)# exit</p>  | Exits adjacency sip configuration mode and enters into SBE configuration mode. |
| Step 7 | <pre>end</pre> <p><b>Example:</b><br/>Router(config-sbc-sbe)# end</p>  | Exits SBE configuration mode and returns to privileged EXEC mode.              |
| Step 8 | <pre>show platform hardware qfp active feature sbc sfx</pre> <p><b>Example:</b><br/>Router# show platform hardware qfp active feature sbc sfx global</p> | Displays the QFP SIP Fast-Register (SFX) counters.                             |

## Information About SoftSwitch Shielding

Cisco Unified Border Element (SP Edition) supports the SoftSwitch Shielding feature that allows a lower SIP registration rate on the links to registrars (typically softswitches) than on the links to endpoints. In a network where endpoints frequently refresh their SIP registrations to a softswitch, allowing a lower registration rate shields the softswitch from an undesirably high rate of re-registrations while ensuring the softswitch still has accurate knowledge of registered endpoints.

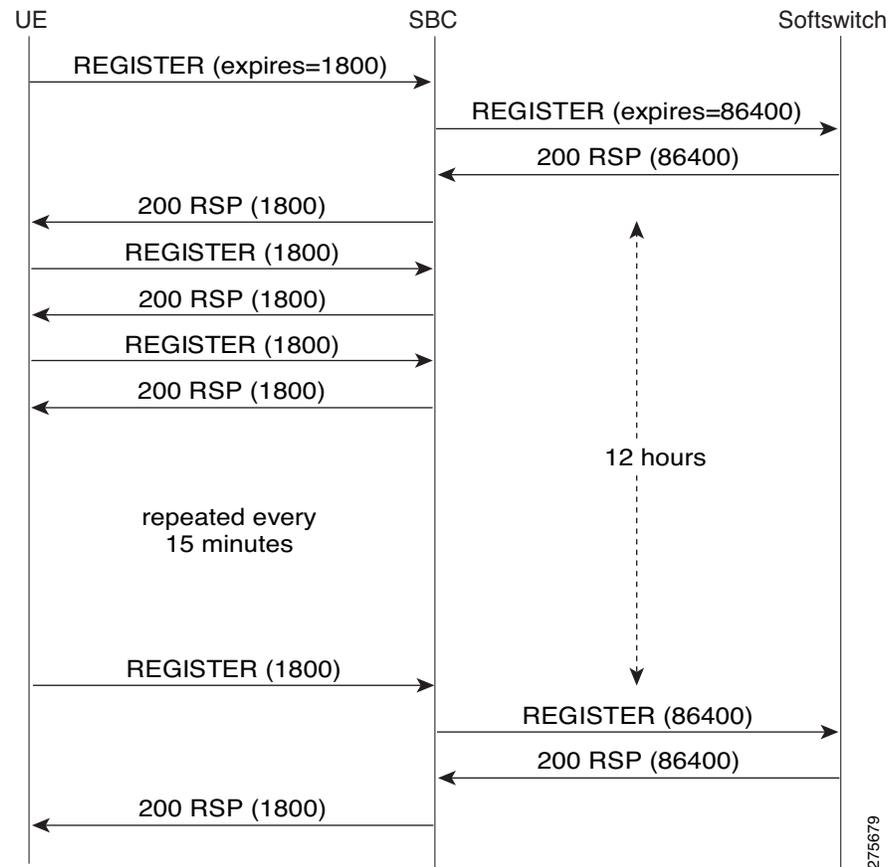
For example, if endpoints are sending REGISTER messages to inform the proxy server of the callee address location every 15 minutes, Cisco Unified Border Element (SP Edition) can be configured to only forward REGISTER messages to the softswitch every 12 hours, unless there is a change to the contact being registered. Using SoftSwitch Shielding reduces the load on the softswitch and the network. On the other hand, if an endpoint stops sending REGISTER messages, Cisco Unified Border Element (SP Edition) detects the change within the endpoint's expiry interval and removes the subscriber state, thus preventing calls to or from this endpoint.

The SoftSwitch Shielding feature gives Cisco Unified Border Element (SP Edition) the capability to shield the softswitch from a large portion of the registration processing. You are also able to simultaneously configure the SIP Fast Registration feature and the SoftSwitch Shielding feature. In addition, if the REGISTER message contains an Authorization header, Cisco Unified Border Element (SP Edition) forwards the REGISTER message to the softswitch registrar.

Use the **registration outgoing timer** command to enable SoftSwitch Shielding and set the time interval during which the SBC forwards REGISTER messages to the softswitch before timing out.

Figure 22-2 illustrates a SoftSwitch Shielding call flow.

Figure 22-2 SoftSwitch Shielding Call Flow



## Configuring SoftSwitch Shielding

This task configures SoftSwitch Shielding on an adjacency.

### SUMMARY STEPS

1. **configure**
2. **sbc *sbc-name***
3. **sbe**
4. **adjacency {sip | h323} *adjacency-name***
5. **registration outgoing timer *sec***
6. **registration rewrite-register**
7. **inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}**

8. **exit**
9. **end**
10. **show sbc *sbc-name* sbe adjacencies *adjacency-name* detail**

## DETAILED STEPS

|        | Command or Action   | Purpose   |
|--------|---|---|
| Step 1 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal  | Enables global configuration mode.  |
| Step 2 | <b>sbc <i>sbc-name</i></b><br><br><b>Example:</b><br>Router(config)# sbc mySbc  | Creates the SBC service on the SBC and enters into SBC configuration mode.  |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe  | Enters the mode of the signaling border element (SBE) function of the SBC.  |
| Step 4 | <b>adjacency {sip   h323} <i>adjacency-name</i></b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip SoftSwitch  | Configures the adjacency facing the registrar, and enters into adjacency sip configuration mode.<br><br><b>Note</b> H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.  |
| Step 5 | <b>registration outgoing timer <i>sec</i></b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# registration outgoing timer 36000   | Enables SoftSwitch Shielding and sets the registration timeout timer for the time interval during which the SBC forwards outgoing REGISTER messages to the softswitch before timing out.<br><br><i>sec</i> —value is 1 to 2147483647. The default of zero disables SoftSwitch Shielding.<br><br>In this example, the time interval is set to every 10 hours or 36000 seconds. |
| Step 6 | <b>registration rewrite-register</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# registration rewrite-register  | Configures the SIP register request rewriting on an adjacency.  |
| Step 7 | <b>inherit profile {preset-access   preset-core   preset-ibcf-ext-untrusted   preset-ibcf-external   preset-ibcf-internal   preset-p-cscf-access   preset-p-cscf-core   preset-peering   preset-standard-non-ims}</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# inherit profile preset-core | Configures a global inherit profile for the SIP adjacency.<br><br>An adjacency facing the registrar typically has a preset-core profile.<br><br>The default is preset-core.   |

|         | Command or Action   | Purpose   |
|---------|---|---|
| Step 8  | <b>exit</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# exit  | Exits adjacency sip configuration mode and enters into SBE configuration mode.  |
| Step 9  | <b>end</b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# end  | Exits SBE configuration mode and returns to Exec mode.  |
| Step 10 | <b>show sbc sbc-name sbe adjacencies adjacency-name detail</b><br><br><b>Example:</b><br>Router# show sbc sbe mySBC sbe adjacencies SoftSwitch detail | Displays all the detailed field output for the specified SIP adjacency, including the “Register Out Timer:” field that shows the time interval in seconds when the SBC forwards the next REGISTER messages to the softswitch. |

## Information About Registration Monitoring

Cisco Unified Border Element (SP Edition) supports creating event subscriptions for changes of registration state. Event subscriptions are generally network-initiated de-registrations. This is a requirement of the IP Multimedia Subsystem (IMS) specifications for Proxy-Call Session Control Function (P-CSCF), a SIP proxy server. For more information, refer to 3rd Generation Partnership Project (3GPP) TS 24.229 v7.5.1.

This support is configured on a per-adjacency basis through the registration monitor field of the Adjacency Table. If this field is set, then Cisco Unified Border Element (SP Edition) creates an event subscription with the registrar for each registered subscriber situated on the adjacency.

The registrar uses the event subscription to provide active indications of changes to the state of the registration. Based on these indications, Cisco Unified Border Element (SP Edition) adds, removes, or updates subscriber state, as appropriate. For more information on event subscriptions, refer to RFC 3680.

Cisco Unified Border Element (SP Edition) does not clean up fast register configuration in the event of a network-initiated de-registration. In this case, the user equipment (UE) is not able to re-register with the registrar until the fast register time period expires.

Cisco Unified Border Element (SP Edition) sets the duration of the monitoring subscription to be the maximum expired interval of the subscriber’s contacts plus a configurable constant. The default monitoring subscription duration is 32 seconds.

Cisco Unified Border Element (SP Edition) only re-subscribes to the Serving-Call Session Control Function’s (S-CSCF) monitoring state when the UE sends a re-register through the SBC. Cisco Unified Border Element (SP Edition) does not follow the 3GPP model of refreshing subscriptions 600 seconds before they expire. The SBC implementation reduces the load on the P-CSCF and S-CSCF, both SIP servers, while ensuring that the subscription lifetime exceeds the registration lifetime, which ensures that network-initiated de-registrations are always detected.

Use the **registration monitor** command to enable monitoring of event subscriptions for registration state changes.

## Configuring Registration Monitoring

This task configures how to monitor event subscriptions as a result of registration state changes.

### SUMMARY STEPS

1. **configure**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency** {**sip** | **h323**} *adjacency-name*
5. **registration monitor**
6. **exit**
7. **end**
8. **show sbc** *sbc-name* **sbe** **adjacencies** *adjacency-name* **detail**

### DETAILED STEPS

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 1 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal   | Enables global configuration mode.   |
| Step 2 | <b>sbc</b> <i>sbc-name</i><br><br><b>Example:</b><br>Router(config)# sbc mySbc   | Creates the SBC service on the SBC and enters into SBC configuration mode.                       |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe   | Enters the mode of the signaling border element (SBE) function of the SBC.                       |
| Step 4 | <b>adjacency</b> { <b>sip</b>   <b>h323</b> } <i>adjacency-name</i><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip<br>Cary-IP-PBX | Configures the adjacency facing the registrar, and enters into adjacency sip configuration mode. |
| Step 5 | <b>registration monitor</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# registration<br>monitor                                      | Enables monitoring of event subscriptions as a result of registration state changes.             |
| Step 6 | <b>exit</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# exit   | Exits adjacency sip configuration mode and enters into SBE configuration mode.                   |

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 7 | <b>end</b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# end   | Exits SBE configuration mode and returns to Exec mode.   |
| Step 8 | <b>show sbc <i>sbc-name</i> sbe adjacencies<br/>adjacency-name detail</b><br><br><b>Example:</b><br>Router# show sbc sbe mySBC sbe adjacencies<br>Cary-IP-PBX detail | Displays all the detailed field output for the specified SIP adjacency, including the “Registration Monitor:” field that shows Registration Monitoring is “Enabled.” |

## Information About Per Subscriber Delete

The Per Subscriber Delete feature provides a mechanism for manually deleting individual subscribers, and associated registered contacts or other subscriber state if any, from the database. This feature works on both manually created subscriber entries and dynamically created entries during the standard registration process. It does not cause the SBC to signal either the subscriber or the registrar; it only removes the internal state from the SBC associated with that subscriber.

Use the **clear sbc sbe sip subscriber aor** command to clear the stuck registrations on Cisco ASR 1000 Series Routers.

## Configuring Per Subscriber Delete

This section shows how to clear stuck registrations.

### SUMMARY STEPS

1. **show sbc *sbc-name* sbe sip subscribers**
2. **clear sbc *sbc-name* sbe sip subscriber aor *address-of-record***

### DETAILED STEPS

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 1 | <b>show sbc <i>sbc-name</i> sbe sip subscribers</b><br><br><b>Example:</b><br>Router# show sbc asr sbe sip subscribers   | Displays details of all the SIP endpoints that have registered with the SBC. Information about the Address of Record (AOR) for each subscriber is also displayed. |
| Step 2 | <b>clear sbc <i>sbc-name</i> sbe sip subscriber aor<br/><i>address-of-record</i></b><br><br><b>Example:</b><br>Router# clear sbc asr sbe sip subscriber aor<br>sip:alice@open-ims.test | Clears the stuck registrations on Cisco ASR 1000 Series Routers.  |

# Information About Aggregate Registration

A registrar is typically a registration server in a SIP network, but outside of the Cisco Unified Border Element (SP Edition) device. The registrar accepts and processes registration requests that register one or more IP addresses to a specific URI, usually a "sip:" address. Because SIP endpoints need to know each others IP address, the registrar acts as a location service. More than one user agent can register at the same IP address. When a call is placed to that IP address, all the registered user agents will ring.

Cisco Unified Border Element (SP Edition) supports Aggregate Registration where a single registration is implemented that causes the registrar to implicitly register multiple IP addresses. The SBC performs aggregate registration for endpoints connected to it. Thus Cisco Unified Border Element (SP Edition) can support devices that implicitly registers multiple endpoints through it. This way of registering all endpoints connected to it in a single registration can be compared to what is commonly done by an IP-PBX device.

The Aggregate Registration feature allows single registration on behalf of multiple endpoints and implicit registration of single endpoints behind the Internet Protocol Private Branch eXchange (IP-PBX).

Aggregate registration is configured on a per-adjacency basis and is configured under an adjacency. All end user clients attached to the adjacency can perform aggregate registration.

When an adjacency has aggregate registration support enabled, the SBC behaves as follows:

- On receiving a REGISTER message, the SBC caches the top Via header and stores it with the normal registration details.
- On receiving an INVITE or out-of-dialog request on the adjacency, Cisco Unified Border Element (SP Edition) attempts to look up the registration using the top Via header, not the Contact and From headers. This ensures that the SBC routes the call to the registrar correctly.
- On receiving an INVITE or out-of-dialog request to the adjacency, Cisco Unified Border Element (SP Edition) overwrites the Request URI as follows:
  - The username is overwritten with the username in the P-Called-Party-Id header if present, or the To header if not.
  - The hostname is overwritten with the hostname that was present on the Contact header that the PBX registered.

Use the **registration aggregate** command to enable Aggregate Registration support from an adjacency.

## Configuring Aggregate Registration

This task configures Aggregate Registration on an adjacency.

### SUMMARY STEPS

1. **configure**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency** {**sip** | **h323**} *adjacency-name*
5. **registration rewrite-register**

6. **inherit profile** {**preset-access** | **preset-core** | **preset-ibcf-ext-untrusted** | **preset-ibcf-external** | **preset-ibcf-internal** | **preset-p-cscf-access** | **preset-p-cscf-core** | **preset-peering** | **preset-standard-non-ims**}
7. **registration aggregate**
8. **header-name** [**contact** [**add** [**tls-param**]] | **from**{**passthrough**} | **to**{**passthrough**}]
9. **request-line request-uri rewrite**
10. **exit**
11. **end**
12. **show sbc** *sbc-name* **sbe** **adjacencies** *adjacency-name* **detail**

## DETAILED STEPS

|        | Command or Action   | Purpose  |
|--------|---|--|
| Step 1 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal  | Enables global configuration mode.   |
| Step 2 | <b>sbc</b> <i>sbc-name</i><br><br><b>Example:</b><br>Router(config)# sbc mySbc  | Creates the SBC service on the SBC and enters into SBC configuration mode.   |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# sbe  | Enters the mode of the signaling border element (SBE) function of the SBC.   |
| Step 4 | <b>adjacency</b> { <b>sip</b>   <b>h323</b> } <i>adjacency-name</i><br><br><b>Example:</b><br>Router(config-sbc-sbe)# adjacency sip<br>Cary-IP-PBX  | Configures the adjacency facing the registrar, and enters into adjacency sip configuration mode.<br><br><b>Note</b> H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier. |
| Step 5 | <b>registration rewrite-register</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# registration<br>rewrite-register   | Configures the SIP register request rewriting on an adjacency.   |
| Step 6 | <b>inherit profile</b> { <b>preset-access</b>   <b>preset-core</b>   <b>preset-ibcf-ext-untrusted</b>   <b>preset-ibcf-external</b>   <b>preset-ibcf-internal</b>   <b>preset-p-cscf-access</b>   <b>preset-p-cscf-core</b>   <b>preset-peering</b>   <b>preset-standard-non-ims</b> }<br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# inherit profile<br>preset-access | Configures a global inherit profile for a SIP adjacency.   |

|         | Command or Action  | Purpose   |
|---------|--|---|
| Step 7  | <b>registration aggregate</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# registration aggregate   | Enables Aggregate Registration support from the adjacency.<br><br><b>Note</b> This step and the next two steps in the correct order (header-name and request-line request-uri rewrite) are required to enable aggregate registration call routing to work completely. |
| Step 8  | <b>header-name [contact [add [tls-param]]   from {passthrough}  to {passthrough}]</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# header-name to passthrough | Configures the passthrough header for non-REGISTER requests.  |
| Step 9  | <b>request-line request-uri rewrite</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# request-line request-uri rewrite   | Allows outgoing calls to the endpoint registered with Aggregate Registration. The SBC rewrites the Request-URI as <user>@<hostname>, before sending a request to the registered subscriber (IP-PBX) on this adjacency.  |
| Step 10 | <b>exit</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-adj-sip)# exit   | Exits adjacency sip configuration mode and enters into SBE configuration mode.  |
| Step 11 | <b>end</b><br><br><b>Example:</b><br>Router(config-sbc-sbe)# end   | Exits SBE configuration mode and returns to Exec mode.  |
| Step 12 | <b>show sbc sbc-name sbe adjacencies adjacency-name detail</b><br><br><b>Example:</b><br>Router# show sbc sbe mySBC sbe adjacencies Cary-IP-PBX detail                     | Displays all the detailed field output for the specified SIP adjacency, including the “Register Aggregate:” field that shows Aggregate Registration is “Enabled.”   |

## Information About Provisioned Delegate Registration

In a SIP network, some third-party client or end user devices are unable to register themselves with a registrar. A registrar is a server that resides outside the SBC device. These end users or clients are generally the applications running on systems used by people. The application may be a softphone application running on your PC or a messaging device in your IP phone. For example, the softphone application generates a request when you try to call another person over the network and sends the request to a server. The generated request is a register message or registration.

End users register their locations or addresses to a registrar server. By registering or sending a special message to a registrar server, the registrar server maintains updated locations of end users in a SIP network. The client then sends the request to a proxy server because when the request is generated, the address of the recipient or callee is not known. The registration is sent to inform a proxy server of the location of the callee address.

Cisco Unified Border Element (SP Edition) can be configured to register any client devices that cannot register themselves. Using the Provisioned Delegate Registration feature, you can set up delegate registration for individual client devices. The client device can make and receive calls as though it had registered normally. Additionally, you can specify individual parameters for each client device, such as registering the client device to a specified registrar server.

Provisioned Delegate Registration is done by provisioning Cisco Unified Border Element (SP Edition) with enough information about a client device so that it can originate a registration for the device itself. Cisco Unified Border Element (SP Edition) can perform end user registration for up to several hundred to a thousand end user clients or delegate clients.

Provisioned Delegate Registration supports the following functionalities:

- Cisco Unified Border Element (SP Edition) can be configured to register with a SIP registrar on behalf of another SIP network entity – the delegate client.
- Users can configure the following information per delegate client:
  - The registrar where the end user client is registered.
  - The registrar facing adjacency if the user wants to bypass normal routing.
  - The adjacency facing the delegate client.
  - The expiration time of the registration.
  - The refresh time of the registration.
  - The contact Uniform Resource Identifier (URI) information of the delegate client. Depending on the adjacency configuration, the URI may be rewritten on messages going to the registrar to force calls to the delegate client to be routed through the SBC.
  - The next hop to the delegate client (for calls coming from the registrar).
  - The address of record (AoR) of the delegate client.
  - The number of attempts and frequency of registration retries when a failure is received.
- Calls from the delegate client can be forwarded to one of the following (in order of preference):
  - the entity identified in the Service-Route header if present on the registration response, or
  - the registrar.
- Calls to the delegate client must have the Request URI rewritten to indicate the delegate client, and forwarded out of the adjacency facing it.

## Restrictions

The following is a restriction of the Provisioned Delegate Registration feature:

- The delegate registration configuration is limited to no more than 1000 subscribers with each subscriber having no more than 5 contacts.

## Provisioned Delegate Registration Call Flow Description

When an end user client is brought up, Cisco Unified Border Element (SP Edition) builds a REGISTER message on behalf of the client, and sends it to the specified registrar. The REGISTER message contains all the contact Uniform Resource Identifier (URI) information that have been configured on the end user client.

If the registrar responds positively to the REGISTER message, Cisco Unified Border Element (SP Edition) stores this fact. Calls to and from the end user client is treated by the SBC exactly as if the client has registered itself.

If the registrar responds negatively to the REGISTER message. For example, if Cisco Unified Border Element (SP Edition) receives error response 423—Interval too brief, the SBC retries building a REGISTER message after the configured retry interval. Cisco Unified Border Element (SP Edition) repeats this process the configured number of times. If the end user client has still failed to be registered, a log is made and the subscriber operating status is changed to OPER\_ACT\_FAILED.

Before the registration time of the end user client expires, Cisco Unified Border Element (SP Edition) performs the registration processing again to refresh the callee/recipient address location through registration.

See [?\\$paranum>Provisional Delegate Registration Commands? section on page 22-26](#) for more information on configuration steps and commands.

## Configuring Delegate Registration Profile

Cisco Unified Border Element (SP Edition) requires a configured subscriber for each end user client upon whose behalf the SBC performs registration. The user may configure retry counts, retry intervals, duration, and the refresh buffer time for each configured subscriber, also called “delegated subscriber.” Several subscribers may all share the same nondefault values for the fields in the Delegate Registrations (amb\_mw\_sudb\_subscriber) table. Instead of requiring configuration for each subscriber separately, Cisco Unified Border Element (SP Edition) allows the user to configure a subscriber profile that can be applied to one or more subscribers.

Use the **delegate-profile** *profile-name* command to configure a profile for a delegate registration subscriber.

## Provisional Delegate Registration Commands

When the AdminStatus of the Delegate Registrations table is set to AdminStatusUp, Cisco Unified Border Element (SP Edition) attempts to register with the registrar. If the registration is successful, the delegate/client device is treated the same as all other subscribers. Cisco Unified Border Element (SP Edition) registers the device for the length of time specified. Cisco Unified Border Element (SP Edition) renews the registration before it expires, with the specified configurable buffer time.

If a registration (or re-registration) fails, Cisco Unified Border Element (SP Edition) retries registration after the configured delay time. Cisco Unified Border Element (SP Edition) retries the specified number of times. If registration still fails, Cisco Unified Border Element (SP Edition) logs the failure and sets the subscriber operating status to failed.

The following commands are used to configure Provisional Delegate Registration:

- Use the **delegate-profile** command to configure a delegate client registration profile that can be applied to a delegate subscriber. After a delegate profile is configured, profile parameters that specify **duration**, **retry-count**, **retry-interval**, and **refresh-buffer** may optionally be configured.
- Use the **subscriber aor** command to define the address of record for the subscriber and define the unique subscriber for whom you want to configure delegate registration. The subscriber must have one or more SIP contacts/URIs associated with it.

- Use the **sip-contact** *uri* command to configure a SIP contact URI for a subscriber. The contact information is used to provision Cisco Unified Border Element (SP Edition) with client device information, so the SBC can register the device. For every delegate registration configured with the **delegate-registration** *hostname* command, one or more SIP contacts/URIs must be configured in the SIP Contacts table (*amb\_mw\_sudb\_local\_id*).
- Use the **delegate-registration** *hostname* command to configure a delegate registration for a delegate client.
- Use the **profile** command to apply a delegate registration profile to a delegate registration subscriber.
- Use the **show sbc sbc name sbe sip subscribers** command to display subscribers for whom Provisioned Delegate Registration has been provisioned.
- Use the **show sbc sbc name sbe sip delegate-profiles** command to display subscriber profiles for whom Provisioned Delegate Registration has been configured.

## Configuring Provisional Delegate Registration

This task configures in order: a profile for a delegate registration subscriber; delegate registration for a specified subscriber associated with an individual client device; delegate registration for a specified client/delegate device; and displays subscribers and subscriber profiles for whom delegate registration have been configured.

### SUMMARY STEPS

1. **configure**
2. **sbc** *sbc-name*
3. **sbe**
4. **delegate-profile** *profile name*
5. **duration** *dur time in secs*
6. **retry-count** *#times to retry*
7. **retry-interval** *retry time in secs*
8. **refresh-buffer** *timeout in secs*
9. **exit**
10. **subscriber** *aor*
11. **sip-contact** *uri*
12. **adjacency** *adjacency name*
13. **exit**
14. **delegate registration** *hostname*
15. **adjacency** *adjacency name*
16. **profile** *my-profile*
17. **activate**
18. **end**
19. **show sbc sbc name sbe sip subscribers delegate**
20. **show sbc sbc name sbe sip delegate-profiles**

## DETAILED STEPS

|        | Command or Action   | Purpose  |
|--------|---|--|
| Step 1 | <b>configure</b><br><br><b>Example:</b><br>Router# <b>configure</b>   | Enables global configuration mode.   |
| Step 2 | <b>sbc</b> <i>sbc-name</i><br><br><b>Example:</b><br>Router(config)# <b>sbc</b> mySbc   | Creates the SBC service on the SBC and enters into SBC configuration mode.   |
| Step 3 | <b>sbe</b><br><br><b>Example:</b><br>Router(config-sbc)# <b>sbe</b>   | Enters the mode of the signaling border element (SBE) function of the SBC.   |
| Step 4 | <b>delegate-profile</b> <i>profile name</i><br><br><b>Example:</b><br>Router(config-sbc-sbe)# <b>delegate-profile</b> my<br>profile                     | Configures a delegate/client registration profile that can be applied to a delegate registration subscriber. Enters into subscriber delegate profile configuration mode where profile parameters can be configured.<br><br>The profile name is a string of 24 characters maximum length. |
| Step 5 | <b>duration</b> <i>dur time in secs</i><br><br><b>Example:</b><br>Router(config-sbc-sbe-subscriber-delegate-prof)<br># <b>duration</b> 100              | Configures the expiration time when the delegate client is due to expire, that is, the length of time in seconds during which the SBC tries to perform delegate registration before stopping.<br><br>The default duration time is 1800 seconds. The range is 1 to 2,147,483 seconds.     |
| Step 6 | <b>retry-count</b> <i>#times to retry</i><br><br><b>Example:</b><br>Router(config-sbc-sbe-subscriber-delegate-prof)<br># <b>retry-count</b> 5           | Configures the number of times the SBC repeats the delegate registration processing after the retry interval ends.<br><br>The default is 3 times. The range is 0 to 255 times.   |
| Step 7 | <b>retry-interval</b> <i>retry time in secs</i><br><br><b>Example:</b><br>Router(config-sbc-sbe-subscriber-delegate-prof)<br># <b>retry-interval</b> 60 | Configures the length of time the SBC waits before it retries delegate registration.<br><br>The default is 30 seconds. The range is 1 to 2,147,483 seconds.  |
| Step 8 | <b>refresh-buffer</b> <i>timeout in secs</i><br><br><b>Example:</b><br>Router(config-sbc-sbe-subscriber-delegate-prof)<br># <b>refresh-buffer</b> 200   | Configures the length of time by which the SBC attempts to renew or refresh the address location with a delegate registration before the specified expiration time ( <b>duration</b> ).<br><br>The default is 30 seconds. The range is 1 to 2,147,483 seconds.                           |
| Step 9 | <b>exit</b><br><br><b>Example:</b><br>Router(config-sbc-sbe-del-prof)# <b>exit</b>  | Exits Subscriber Delegate Profile configuration mode and enters SBE configuration mode.  |

|         | Command or Action   | Purpose  |
|---------|---|--|
| Step 10 | <p><b>subscriber</b> <i>aor</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe)# subscriber<br/> sip:bob@isp.example</p>   | Configures a delegate registration for a specified subscriber associated with an individual client device. Enters into subscriber-entry configuration mode where SIP contact info can be configured for the delegate registration.                           |
| Step 11 | <p><b>sip-contact</b> <i>uri</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-entry)#<br/> sip-contact sip:steve@10.1.1.2</p>                             | <p>Configures the SIP contact information for a specified URI IP address location or address of record. Enters into subscriber-contact (SIP) configuration mode.</p> <p>The URI is a string of 62 maximum character length.</p>                              |
| Step 12 | <p><b>adjacency</b> <i>adjacency name</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-contact)#<br/> adjacency CallMgrB</p>                              | Configures the mandatory local subscriber adjacency name of the configured SIP contact.  |
| Step 13 | <p><b>exit</b></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-contact)# exit</p>  | Exits subscriber-contact (SIP) configuration mode and enters into subscriber-entry configuration mode to configure delegate registration.  |
| Step 14 | <p><b>delegate-registration</b> <i>hostname</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-entry)#<br/> delegate-registration sip:registrar@1.1.1.1</p> | <p>Configures a delegate registration for a specified client device or delegate client, and enters into subscriber-delegate configuration mode where registration parameters can be set.</p> <p>The hostname is a string of 64 maximum character length.</p> |
| Step 15 | <p><b>adjacency</b> <i>adjacency name</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-delegate)#<br/> adjacency CallMgrA</p>                             | Configures the adjacency facing the registrar.   |
| Step 16 | <p><b>profile</b> <i>profile name</i></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-delegate)#<br/> profile my profile</p>                                 | Applies the delegate registration profile, created previously with the <b>delegate-profile</b> command, to a delegate registration subscriber.   |
| Step 17 | <p><b>activate</b></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-delegate)#<br/> activate</p>  | (Required) Activates the delegate registration.  |
| Step 18 | <p><b>end</b></p> <p><b>Example:</b><br/> Router(config-sbc-sbe-subscriber-delegate)# end</p>   | Exits subscriber-delegate configuration mode and returns to Privileged EXEC mode.  |

|         | Command or Action  | Purpose   |
|---------|--|---|
| Step 19 | <b>show sbc <i>sbc name</i> sbe sip subscribers delegate</b>           | Displays subscribers for whom delegate registration has been configured. The <b>delegate</b> keyword displays the associated URI contact information for subscribers. |
|         | <b>Example:</b><br>Router# show sbc mySBC sbe sip subscribers delegate |   |
| Step 20 | <b>show sbc <i>sbc name</i> sbe sip delegate-profiles</b>              | Displays subscriber profiles for subscribers for whom delegate registration has been configured.  |
|         | <b>Example:</b><br>Router# show sbc mySBC sbe sip delegate-profiles    |   |

## Configuration Examples

This section has the following configuration examples:

- [SIP Fast Registration Example, page 22-30](#)
- [SoftSwitch Shielding and Aggregate Registration Configuration Examples, page 22-31](#)
- [Registration Monitoring Examples, page 22-35](#)
- [Provisional Delegate Registration Examples, page 22-36](#)
- [Contact Username Passthrough Examples, page 22-38](#)
- [Alternative Contact Rewriting Example, page 22-39](#)
- [Registering with Softswitch via Cisco SRP Integrated Access Device \(IAD\) Examples, page 22-39](#)

## SIP Fast Registration Example

Use the **show platform hardware qfp active feature sbc sfx** command to display the QFP SIP Fast-Register (SFX) counters. Information about how SIP fast-register (SFX) messages are processed, including which SIP REGISTER request packets are punted to the Route Processor (RP) or dropped, may help explain why call rates are low and why the RP CPU load is high.

The following example shows information about the parsing of SIP fast-register (SFX) messages in the Cisco QuantumFlow Processor (QFP):

```
Router# show platform hardware qfp active feature sbc sfx global
```

```
SBC QFP SIP Fast Register Data Plane Information
```

```
-----
SIP 200 OK Replies generated           = 0
SIP REGISTER punts :
  No table entry                       = 0
  Fast Timer expiry                    = 0
  Expires=0                            = 0
  SIP Syntax Error                     = 0
  QFP Out of Resources                  = 0
  QFP Internal Error                    = 0
SIP REGISTER drops :
  QFP Internal Error                   = 0
  UDP Length Error                     = 0
  UDP Checksum Error                   = 0
```

## SoftSwitch Shielding and Aggregate Registration Configuration Examples

The following is a configuration example showing that Aggregate Registration and SoftSwitch Shielding are configured:

```
sbc test
sbe
  sip header-profile myheader
  header P-Called-Party-ID entry 1
  action pass
  adjacency sip sippa =====> Adjacency facing IP-PBX
  header-profile inbound myheader
  header-profile outbound myheader
  inherit profile preset-access
  preferred-transport udp
  signaling-address ipv4 99.99.103.150
  signaling-port 5080
  remote-address ipv4 100.100.1.64 255.255.255.255
  signaling-peer 100.100.1.64
  signaling-peer-port 5080
  registration rewrite-register
  account sipp-a
  registration aggregate
  fast-register disable
  header-name to passthrough
  request-line request-uri rewrite

  attach
  adjacency sip sippb =====> Adjacency facing Registrar
  nat force-off
  header-profile inbound myheader
  header-profile outbound myheader
  inherit profile preset-core
  preferred-transport udp
  signaling-address ipv4 99.99.103.150
  signaling-port 5082
  remote-address ipv4 100.100.1.64 255.255.255.255
  signaling-peer 100.100.1.64
  signaling-peer-port 5082
  account sipp-b
  registration target address 100.100.1.64
  registration target port 5084
  fast-register disable
  attach
  cac-policy-set 1
  first-cac-table mytable
  first-cac-scope src-adjacency
  cac-table mytable
  table-type limit adjacency
  entry 1
  match-value sippa
  max-num-calls 10
  action cac-complete
  complete
  active-cac-policy-set 1
  call-policy-set 1
  first-call-routing-table src-acc-table
  first-reg-routing-table src-acc-table
  rtg-src-adjacency-table src-acc-table
  entry 1
  action complete
  dst-adjacency sippb
  match-adjacency sippa
```

```

    entry 2
      action complete
      dst-adjacency sippa
      match-adjacency sippb
    complete
  call-policy-set 2
  active-call-policy-set 1
!
vdbe global
  unexpected-source-alerting
  media-address ipv4 99.99.103.156
  media-timeout 9999
  activate
!
Softswitch shielding config
=====
sbc test
  sbe
    adjacency sip sippa
      signaling-address ipv4 99.99.103.150
      signaling-port 5080
      remote-address ipv4 100.100.1.64 255.255.255.255
      signaling-peer 100.100.1.64
      signaling-peer-port 5080
      registration rewrite-register
      account sipp-a
      attach
    adjacency sip sippb
      signaling-address ipv4 99.99.103.150
      signaling-port 5082
      remote-address ipv4 100.100.1.64 255.255.255.255
      signaling-peer 100.100.1.64
      signaling-peer-port 5082
      account sipp-b
      registration outgoing timer 86400
      registration target address 100.100.1.64
      registration target port 5084
      attach
  call-policy-set 1
    first-call-routing-table src-acc-table
    first-reg-routing-table src-acc-table
    rtg-src-adjacency-table src-acc-table
  entry 1
    action complete
    dst-adjacency sippb
    match-adjacency sippa
  entry 2
    action complete
    dst-adjacency sippa
    match-adjacency sippb
  complete
  active-call-policy-set 1
!
  media-address ipv4 99.99.103.156
  media-timeout 9999
  activate
!

```

```
Router# show sbc test sbe adjacencies sippb detail
```

```
SBC Service "test"
Adjacency sippb (SIP)
  Status:                Attached
  Signaling address:     99.99.103.150:5082
```

```

Signaling-peer:      100.100.1.64:5082
Force next hop:     No
Account:            sipp-b
Group:              None
In header profile:  Default
Out header profile: Default
In method profile:  Default
Out method profile: Default
In UA option prof:  Default
Out UA option prof: Default
In proxy opt prof:  Default
Out proxy opt prof: Default
Priority set name:   None
Local-id:           None
Rewrite REGISTER:   Off
Target address:     100.100.1.64:5084
NAT Status:         Auto Detect
Reg-min-expiry:     3000 seconds
Fast-register:      Enabled
Fast-register-int:  30 seconds
Register aggregate: Disabled
Register Out Interval: 86400 seconds
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID:  None
Inherit profile:    Default
Force next hop:     No
Home network Id:    None
UnEncrypt key data: None
SIPI passthrough:   No
Rewrite from domain: Yes
Rewrite to header:  Yes
Media passthrough:  No
Client authentication: No
Hunting Triggers:   Global Triggers
Redirect mode:       Pass-through
Security:            Untrusted-Unencrypted
Signaling Peer Status: Not Tested
Rewrite Request-uri: Disabled
Registration Monitor: Disabled
DTMF SIP NOTIFY Relay: Enabled
DTMF SIP NOTIFY Interval: 2000
DTMF SIP default duration: 200
DTMF Preferred Method: SIP NOTIFY

```

The following example configures SoftSwitch Shielding on adjacency “SoftSwitch:”

```

sbc mySbc
sbe
  adjacency sip SoftSwitch
  registration outgoing timer <sec>
  registration rewrite-register
  inherit profile preset-core

```

The following example shows detailed output for adjacency SoftSwitch, including the “Register Out Timer:” field that shows the time interval in seconds when the SBC forwards the next REGISTER messages to the softswitch.

```

Router# show sbc mySbc sbe adjacencies SoftSwitch detail

SBC Service "mySbc"
  Adjacency SoftSwitch (SIP)
    Status:                Attached

```

```

Signaling address:      100.100.100.100:5060, VRF Admin
Signaling-peer:        10.10.51.10:5060
Force next hop:        No
Account:
Group:                 None
In header profile:     Default
Out header profile:    Default
In method profile:     Default
Out method profile:    Default
In UA option prof:     Default
Out UA option prof:    Default
In proxy opt prof:     Default
Out proxy opt prof:    Default
Priority set name:      None
Local-id:              None
Rewrite REGISTER:      Off
Target address:        None
Register Out Timer:    36000 seconds
Register Aggregate:    Disabled
NAT Status:            Auto Detect
Reg-min-expiry:        30 seconds
Fast-register:         Enabled
Fast-register-int:     30 seconds
Authenticated mode:    None
Authenticated realm:   None
Auth. nonce life time: 300 seconds
IMS visited NetID:     None
Inherit profile:       Default
Force next hop:        No
Home network Id:       None
UnEncrypt key data:    None
SIPI passthrough:     No
Rewrite from domain:   Yes
Rewrite to header:     Yes
Media passthrough:     No
Preferred transport:   UDP
Hunting Triggers:      Global Triggers
Redirect mode:          Pass-through
Security:              Untrusted
Outbound-flood-rate:   None
Ping-enabled:          No
Signaling Peer Status: Not Tested

```

The following example enables Aggregate Registration on adjacency Cary-IP-PBX, which has a preset access profile specified because it faces an access device on a UNI network. The last three commands in the configuration, entered in the correct order, enable the aggregate registration call routing to work.

```

sbc mySbc
sbe
adjacency sip Cary-IP-PBX
registration rewrite-register
inherit profile preset-access
registration aggregate
header-name to passthrough
request-line request-uri rewrite

```

The following example displays detailed output for adjacency Cary-IP-PBX, including the “Register Aggregate:” field that shows Aggregate Registration is “Enabled.”

```
Router# show sbc mySbc sbe adjacencies Cary-IP-PBX detail
```

```

SBC Service "mySBC"
Adjacency Cary-IP-PBX (SIP)
Status:                Attached

```

```

Signaling address:      100.100.100.100:5060, VRF Admin
Signaling-peer:        10.10.51.10:5060
Force next hop:        No
Account:
Group:                 None
In header profile:     Default
Out header profile:    Default
In method profile:     Default
Out method profile:    Default
In UA option prof:     Default
Out UA option prof:    Default
In proxy opt prof:     Default
Out proxy opt prof:    Default
Priority set name:      None
Local-id:              None
Rewrite REGISTER:      Off
Target address:        None
Register Out Timer:    1800 seconds
Register Aggregate:   Enabled
NAT Status:            Auto Detect
Reg-min-expiry:        30 seconds
Fast-register:         Enabled
Fast-register-int:     30 seconds
Authenticated mode:    None
Authenticated realm:   None
Auth. nonce life time: 300 seconds
IMS visited NetID:     None
Inherit profile:       Default
Force next hop:        No
Home network Id:       None
UnEncrypt key data:    None
SIP passthrough:       No
Rewrite from domain:   Yes
Rewrite to header:     Yes
Media passthrough:     No
Preferred transport:   UDP
Hunting Triggers:      Global Triggers
Redirect mode:          Pass-through
Security:              Untrusted
Outbound-flood-rate:   None
Ping-enabled:          No
Signaling Peer Status: Not Tested
Rewrite Request-uri:  Enabled
Registration Monitor:   Disabled

```

## Registration Monitoring Examples

The following example shows how monitoring of event subscriptions as a result of registration state changes is enabled:

```

sbc Raleigh-SBC
sbe
 adjacency sip Cary-IP-PBX
 registration monitor

```

The following example displays detailed output for adjacency Cary-IP-PBX, including the “Registration Monitor:” field that shows Registration Monitoring is “Enabled:”

```

Router# show sbc mySBC sbe adjacencies Cary-IP-PBX detail

SBC Service "mySbc"
Adjacency Cary-IP-PBX (SIP)

```

```

Status: Attached
Signaling address: 100.100.100.100:5060, VRF Admin
Signaling-peer: 10.10.51.10:5060
Force next hop: No
Account:
Group: None
In header profile: Default
Out header profile: Default
In method profile: Default
Out method profile: Default
In UA option prof: Default
Out UA option prof: Default
In proxy opt prof: Default
Out proxy opt prof: Default
Priority set name: None
Local-id: None
Rewrite REGISTER: Off
Target address: None
Register Out Timer: 1800 seconds
Register Aggregate: Enabled
NAT Status: Auto Detect
Reg-min-expiry: 30 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID: None
Inherit profile: Default
Force next hop: No
Home network Id: None
UnEncrypt key data: None
SIPi passthrough: No
Rewrite from domain: Yes
Rewrite to header: Yes
Media passthrough: No
Preferred transport: UDP
Hunting Triggers: Global Triggers
Redirect mode: Pass-through
Security: Untrusted
Outbound-flood-rate: None
Ping-enabled: No
Signaling Peer Status: Not Tested
Rewrite Request-uri: Disabled
Registration Monitor: Enabled

```

## Provisional Delegate Registration Examples

The following example configures a delegate registration profile that can be applied to a delegate registration subscriber.

```

sbc mySbc sbe
  delegate-profile my-profile
  duration 1000
  retry-count 5
  retry-interval 60
  refresh-buffer 200

```

The following example configures a SIP contact for a subscriber, for whom a subscriber detail table exists, and for whom, after the SIP contact is configured, delegate registration can be configured:

```

sbc mySbc

```

```
sbe
subscriber sip:bob@isp.example
sip-contact sip:steve@10.1.1.2
adjacency CallMgrB
exit
```

The following example configures a delegate registration for a specified client device address location, after the SIP contact information has been configured:

```
sbc mySbc
sbe
subscriber sip:bob@isp.example
sip-contact sip:steve@10.1.1.2
adjacency CallMgrB                =====> client adjacency
exit
delegate-registration sip:registrar@1.1.1.1
adjacency CallMgrA                =====> registrar adjacency
activate
```

The following show example displays subscribers for which delegate registration have been configured. The **delegate** keyword displays the associated URI contact information for subscribers.

```
Router# show sbc mySBC sbe sip subscribers delegate
```

```
0          1          2          3 4          5          6          7
012345789012345789012345789012345789012345789012345789012345789012345789
```

```
AOR:                sip:steve1.cisco.com
Subscriber Location[s]: sip:contact@cisco.com -> CallMgrC
                    sip:contact2@cisco.com -> CallMgrD
Registrar adj:      CallMgrA
Registrar:          sip:myreg@172.18.52.148
Register Duration:  1800
Register Retries:   3
Retry Interval:     30
Refresh Buffer:      30
Time left:          0 days
```

The following show example displays subscriber profiles for subscribers for whom delegate registration has been configured.

```
Router# show sbc mySBC sbe sip delegate-profiles
```

```
0          1          2          3 4          5          6          7
012345789012345789012345789012345789012345789012345789012345789012345789
```

```
Delegate Profiles:
```

```
-----
Profile                = steve
Duration (secs)        = 1800
Retry Count             = 3
Retry Interval (secs)  = 30
Refresh Buffer (secs)   = 30
-----
```

## Contact Username Passthrough Examples

The following is an example with a single contact showing that the username part of the contact is passed through unchanged:

```
adjacency sip SIPPlReg
  group SIPPlReg
  inherit profile preset-core
  signaling-address ipv4 192.168.101.1
  statistics-setting summary
  signaling-port 5060 5062
  remote-address ipv4 192.168.101.12 255.255.255.255
  signaling-peer 192.168.101.12
  signaling-peer-port 7068
  registration target address 192.168.101.12
  registration target port 7069
  registration contact username passthrough
attach
```

The following is an example flow of multiple registrations for the same subscriber; the example illustrates how a sequence of REGISTER requests registering multiple contacts behaves. This example assumes all headers are omitted from the requests, apart from Contact headers, and that the registrar-facing adjacency has a signaling-port range from 5060 to 5063 (this means 4 local ports are available).

- 
- Step 1** A REGISTER is received registering two contact addresses for the number 5551234.
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@1.1.1.1>
Contact: <sip:robert@1.1.1.1>
```
- Step 2** The SBC forwards this REGISTER to the registrar having rewritten the contact address and port.
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@192.168.101.1:5060>
Contact: <sip:robert@192.168.101.1:5061>
```
- Step 3** Another REGISTER is received for the number 5551234, registering another endpoint with a duplicate username of “bob.”
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@2.2.2.2>
```
- Step 4** The SBC forwards this to the registrar, again passing the username through unchanged.
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@192.168.101.1:5062>
```
- Step 5** A third endpoint registers for the same number. This endpoint provides a very long contact name in the contact field.
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:this_is_an_extremely_long_contact_username@2.2.2.2>
```
- Step 6** The SBC forwards this request to the registrar and rewrites the username because it is over the maximum passthrough length (32).
- ```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip: 6e83bca53a48bd629a153a93ff8f4af1@192.168.101.1:5063>
```

## Alternative Contact Rewriting Example

The following example shows how to configure the Alternative Contact Rewriting feature on the SBC:

```
sbc test
sbe
  adjacency sip core-side-1
    force-signaling-peer
    nat force-off
    inherit profile preset-core
    signaling-address ipv4 9.9.9.1
    remote-address ipv4 10.0.49.78 255.255.255.255
    signaling-peer 10.0.49.78
    registration target address 10.0.49.78
  registration target port 5060
  registration contact username rewrite userid-and-numeric
  attach

  adjacency sip core-side-2
    force-signaling-peer
    nat force-off
    inherit profile preset-core
    signaling-address ipv4 9.9.9.2
    remote-address ipv4 10.0.49.76 255.255.255.255
    signaling-peer 10.0.49.76
    registration target address 10.0.49.76
  registration target port 5060
  registration contact username rewrite numeric
  attach
```

## Registering with Softswitch via Cisco SRP Integrated Access Device (IAD) Examples

The following example shows how to register a subscriber with Softswitch, using the Cisco SRP Integrated Access Device (IAD).

This example uses IP realms in the adjacency. See [IP Realm Support](#) for information on IP Realms.

```
sbc interop
sbe
  adjacency sip srp
    nat force-on
    inherit profile preset-access
    preferred-transport udp
    signaling-address ipv4 10.3.127.1
    statistics method summary
    signaling-peer 0.0.0.0
    registration rewrite-register
    realm customer.com
  attach
  adjacency sip meta
    nat force-off
    inherit profile preset-core
    preferred-transport udp
    signaling-address ipv4 99.109.206.106
    statistics method summary
    signaling-peer sig.trav.demo.softswitch.com
    registration target address sig.trav.demo.softswitch.com
    registration target port 5060
```

```
fast-register disable
header-name To passthrough
header-name From passthrough
realm sig.trav.demo.softswitch.com
attach
call-policy-set 1
first-call-routing-table crtab1
first-reg-routing-table crtab1
rtg-src-adjacency-table crtab1
  entry 1
    action complete
    dst-adjacency meta
    match-adjacency srp
  entry 2
    action complete
    dst-adjacency srp
    match-adjacency meta
  complete
active-call-policy-set 1
!
!
!
media-address ipv4 10.3.127.1 realm customer.com
media-address ipv4 99.109.206.106 realm sig.trav.demo.softswitch.com
activate
!
```