



SUBSCRIBE-NOTIFY Passthrough

- [Overview, on page 1](#)
- [Restrictions, on page 3](#)
- [Information About SUBSCRIBE-NOTIFY Passthrough, on page 3](#)
- [Configure SUBSCRIBE-NOTIFY Passthrough, on page 5](#)
- [Configuration Examples for SUBSCRIBE-NOTIFY Passthrough, on page 10](#)

Overview

The SUBSCRIBE-NOTIFY mechanism is used for implementation of features such as Message Waiting Indication (MWI), Shared Call Appearance, Multiple Caller Appearance, Busy Lamp Field, and so on.

In CUBE, the SUBSCRIBE-NOTIFY framework on Unified Communications (UC) products supports the following:

- Configurable and Selective Passthrough of SUBSCRIBE and NOTIFY transactions from phones with the normalization that is required for address or topology hiding and dialog content updates for “dialog” event subscription.
- Survivability mode handling of incoming SUBSCRIBE request for critical events.

The key attributes of the SUBSCRIBE-NOTIFY Passthrough feature are as follows:

- Message Passthrough Application (MPA)—The SIP MPA handles SUBSCRIBE-NOTIFY passthrough. This application maintains subscribe dialogs, and references the dial-peer database and registration passthrough configurations to route the initial SUBSCRIBE and unsolicited NOTIFY requests.
- Header Passthrough—All the non-mandatory headers in SUBSCRIBE-NOTIFY requests and responses are passed through from one endpoint to the other.
- Content Passthrough—The content bodies in SUBSCRIBE-NOTIFY requests are passed through transparently from one endpoint to the other.
- “Dialog” Event Content Manipulation—The content in the NOTIFY body for a dialog event is updated before passthrough when the dialog is maintained by the CUBE.
- Passthrough Configuration and Filtering—SUBSCRIBE-NOTIFY passthrough is configurable globally as well as under dial-peer, and can be configured for selected events using the configuration of an event list.

- **Error Passthrough for SUBSCRIBE-NOTIFY Requests**—When an error is received for a SUBSCRIBE-NOTIFY request, the error is passed through to the peer with the relevant headers.
- **Backward Compatibility**—The SIP MPA has the highest priority when SUBSCRIBE-NOTIFY passthrough is enabled. If passthrough is not enabled (either for all events or for a specific event), the current applications will control the incoming requests and responses.
- **401/407 Error Message Passthrough**—SIP message 401/407 is sent by the user-agent server (UAS) or end device to challenge messages like INVITE/REFER/SUBSCRIBE and request for endpoint credentials information. CUBE does not store endpoint credential information to act on behalf of phone or endpoint. To enable the passthrough of 401/407, you can enable the **error passthru** command at the global level. The messages 401/407 are in passthru mode for INVITE/REFER/SUBSCRIBE.

Feature Information

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for SUBSCRIBE-NOTIFY Passthrough

Feature Name	Releases	Feature Information
CUBE (SUBSCRIBE-NOTIFY Passthrough)	Cisco IOS XE Fuji 16.9.1	<p>The SUBSCRIBE-NOTIFY mechanism is used for implementation of features such as Message Waiting Indication (MWI), Shared Call Appearance, Multiple Caller Appearance, Busy Lamp Field, and so on.</p> <p>In CUBE, the SUBSCRIBE-NOTIFY framework on Unified Communications (UC) products supports the following:</p> <ul style="list-style-type: none"> • Configurable and Selective Passthrough of SUBSCRIBE and NOTIFY transactions from phones with the normalization required for address or topology hiding and dialog content updates for “dialog” event subscription. • Survivability mode handling of incoming SUBSCRIBE request for critical events.

Restrictions

- The SUBSCRIBE-NOTIFY passthrough framework can only pass through events when there is a one-to-one association between the incoming request and the outgoing location to which the request has been sent out. This means that either:
 - There should be an outbound dial-peer identified for the request received.
 - or
 - The outbound target for the request could be only a single registrar.
- The following use cases are not supported:
 - SUBSCRIBE-NOTIFY passthrough with hunting of outbound dial-peers to which the subscribe or notify requests need to be sent.
 - SUBSCRIBE-NOTIFY passthrough where an inbound dial-peer has peer-to-peer mode of registration passthrough enabled with more than one registrar (there will be no forking of Subscribe-Notify Requests)
 - Local subscribe handling of unsupported events when the remote registrar is unavailable. Local subscribe handling is only applicable to cases where the inbound dial-peer matching the subscribe has registration passthrough enabled with “local-fallback.”

Information About SUBSCRIBE-NOTIFY Passthrough

The key attributes of the SUBSCRIBE-NOTIFY Passthrough feature are as follows:

- **Message Passthrough Application (MPA)**—The SIP MPA handles SUBSCRIBE-NOTIFY passthrough. This application maintains subscribe dialogs, and references the dial-peer database and registration passthrough configurations to route the initial SUBSCRIBE and unsolicited NOTIFY requests.
- **Header Passthrough**—All the non-mandatory headers in SUBSCRIBE-NOTIFY requests and responses are passed through from one endpoint to the other.
- **Content Passthrough**—The content bodies in SUBSCRIBE-NOTIFY requests are passed through transparently from one endpoint to the other.
- **“Dialog” Event Content Manipulation**—The content in the NOTIFY body for a dialog event is updated before passthrough when the dialog is maintained by the CUBE.
- **Passthrough Configuration and Filtering**—SUBSCRIBE-NOTIFY passthrough is configurable globally as well as under dial-peer, and can be configured for selected events using the configuration of an event list.
- **Error Passthrough for SUBSCRIBE-NOTIFY Requests**—When an error is received for a SUBSCRIBE-NOTIFY request, the error is passed through to the peer with the relevant headers.
- **Backward Compatibility**—The SIP MPA has the highest priority when SUBSCRIBE-NOTIFY passthrough is enabled. If passthrough is not enabled (either for all events or for a specific event), the current applications will control the incoming requests and responses.

- 401/407 Error Message Passthrough—SIP message 401/407 is sent by the user-agent server (UAS) or end device to challenge messages like INVITE/REFER/SUBSCRIBE and request for endpoint credentials information. CUBE does not store endpoint credential information to act on behalf of phone or endpoint. To enable the passthrough of 401/407, you can enable the **error passthru** command at the global level. The messages 401/407 are in passthru mode for INVITE/REFER/SUBSCRIBE.

SUBSCRIBE-NOTIFY Passthrough Request Routing

The first step of request or response routing is for Cisco Unified Border Element (CUBE) to determine whether or not the request has to be passed through. When a new SUBSCRIBE or unsolicited NOTIFY request arrives, its headers are used to match an incoming dial-peer. If the incoming dial-peer has SUBSCRIBE-NOTIFY Passthrough (SNPT) enabled or if there is no incoming dial-peer and global SNPT is enabled for that event, then the request is handed off to be passed through. For solicited subscriptions, the passthrough check is applicable only to the initial SUBSCRIBE request; subsequent requests or responses are not checked and will be routed based on updated dialog parameters.

The second step is to determine the outbound destination of the SUBSCRIBE or unsolicited NOTIFY request.

- **First Pass: Outbound dial-peer match**—An outbound VoIP dial-peer is first matched based on the request headers (From, To, and Via), the Subscriber Number (userid in the To header), and the incoming dial-peer Class of Restrictions (CoR) if any. If there is a match, the request is routed to the session target.
- **Second Pass: Configured registrar for registration passthrough in peer-to-peer mode**—If no outbound dial-peer is found and the incoming dial-peer has registration passthrough enabled in static (peer-to-peer) mode with a single registrar configured, then the request is routed to the registrar address.
- **Third Pass: Configured registrar for registration passthrough in end-to-end mode**—If no outbound dial-peer is found and the incoming dial-peer has registration passthrough enabled in dynamic (end-to-end) mode:
 - If the request Uniform Resource Identifier (URI) has the CUBE IP address, the request is routed to the configured registrar if only a single registrar is configured.
 - If the request URI has a non-CUBE IP address, then the request is routed to that IP address.
- **Fourth Pass: Request URI-based routing**—If no outbound dial-peer is found and no registration passthrough is configured, the request URI is used to route the request if it does not point to the CUBE's IP address.

SUBSCRIBE-NOTIFY Passthrough Survivability Mode

In survivability mode, the CUBE could encounter the following scenarios:

- When the CUBE receives a line-seize (event) subscribe in survivability mode, it checks the line-seize queue to see if another phone has already seized the same line; if not, CUBE accepts the subscription, sends a NOTIFY response with State = Active, and starts the timer for expiration. In survivability mode, SUBSCRIBE received for any event other than line-seize is rejected.
- If another phone has already subscribed for the line, CUBE sends a 200 OK (request successful) response for the new subscribe, but a final NOTIFY to indicate that the subscription has been terminated.
- If the subscription timer expires without re-subscription from the phone, CUBE sends a final NOTIFY to remove the subscription.

- If a subscription is created in active mode, but re-subscriptions or unsubscriptions are received in survivability mode, then CUBE returns an error for this subscription.

Configure SUBSCRIBE-NOTIFY Passthrough

Configure an Event List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-event** *number*
4. **event** *name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	voice class sip-event <i>number</i> Example: Device(config)# voice class sip-event 1	Enters voice class configuration mode and configures the list of events to be passed through.
Step 4	event <i>name</i> Example: Device(config-class)# event message-summary	Adds the name of the event to be added to the event list.
Step 5	end Example: Device(config-class)# end	Returns to privileged EXEC mode.

Configure SUBSCRIBE-NOTIFY Event Passthrough Globally

SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. pass-thru subscribe-notify-events *tag*
6. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	voice service voip Example: Device(config)# voice class voip	Enters voice service VoIP configuration mode.
Step 4	sip Example: Device(conf-voi-serv)# sip	Enters voice service SIP configuration mode.
Step 5	pass-thru subscribe-notify-events <i>tag</i> Example: Device(conf-serv-sip)# pass-thru subscribe-notify-events 1	Configures SUBSCRIBE-NOTIFY passthrough event with the SIP event list tag number to be linked globally. <ul style="list-style-type: none"> • You can use the pass-thru subscribe-notify-events all command to configure passthrough for all SUBSCRIBE-NOTIFY events.
Step 6	end Example: Device(conf-serv-sip)# end	Returns to privileged EXEC mode.

Configure SUBSCRIBE-NOTIFY Event Passthrough at the Dial-Peer Level

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `voice-class sip pass-thru subscribe-notify-events tag`
5. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	dial-peer voice tag voip Example: <pre>Device(config)# dial-peer voice 123 voip</pre>	Enters dial peer voice configuration mode.
Step 4	voice-class sip pass-thru subscribe-notify-events tag Example: <pre>Device(config-dial-peer)# voice-class sip pass-thru subscribe-notify-events 1</pre>	Configures SUBSCRIBE-NOTIFY passthrough event with the SIP event list tag number to be linked globally. <ul style="list-style-type: none"> • You can use the voice-class sip pass-thru subscribe-notify-events all command to configure passthrough for all SUBSCRIBE-NOTIFY events.
Step 5	end Example: <pre>Device(conf-serv-sip)# end</pre>	Returns to privileged EXEC mode.

Verify SUBSCRIBE-NOTIFY Passthrough

Perform this task to verify the configuration for SUBSCRIBE-NOTIFY Passthrough and to verify the subscriptions created. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice *number* | inc pass**
3. **show subscription asnl session active**
4. **show subscription sip**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show dial-peer voice *number* | inc pass

Displays the information for voice dial peers. The following sample output shows the configured SUBSCRIBE-NOTIFY passthrough event for a particular dial peer.

Example:

```
Device# show dial-peer voice 123 | inc pass

ip media DSCP = ef, ip media rsvp-pass DSCP = ef
ip video rsvp-none DSCP = af41, ip video rsvp-pass DSCP = af41
voice class sip pass-thru headers = system,
voice class sip pass-thru subscribe-notify-events = system,
voice class sip pass-thru content unsupp = system,
voice class sip pass-thru content sdp = system,
voice class sip privacy-policy passthru = system,
voice class sip registration passthrough = System
voice class sip refer-to-passing = system
```

Step 3 show subscription asnl session active

Displays information about Application Subscribe/Notify Layer (ASNL)-based and non-ASNL-based SIP subscriptions.

Example:

```
Device# show subscription asnl session active
```

```
ASNL Active Subscription Records Details:
=====
Number of active subscriptions: 1
URL: sip:user@10.7.104.88
  Event Name : stress
  Session ID : 8
  Expiration Time : 50 seconds
  Subscription Duration : 5 seconds
  Protocol : ASNL_PROTO_SIP
  Remote IP address : 10.7.104.88
  Port : 5060
  Call ID : 5
  Total Subscriptions Sent : 1
  Total Subscriptions Received: 0
  Total Notifications Sent : 0
```



```

Total Notifications Received : 2
Last response code : ASNL_NOTIFY_RCVD
Last error code : ASNL_NONE
First Subscription Time : 10:55:12 UTC Apr 9 2000
Last Subscription Time : 10:55:12 UTC Apr 9 2000
First Notify Time : 10:55:12 UTC Apr 9 2000
Last Notify Time : 10:55:17 UTC Apr 9 2000
Application that subscribed : stress
Application receiving notification: stress

```

Step 4 show subscription sip

Displays information about ASNL-based and non-ASNL-based SIP subscriptions.

Example:

```
Device# show subscription sip
```

```
ASNL Active Subscription Records Summary:
=====
```

```
Number of active subscriptions: 2
```

SubId	CallId	Proto	URL	Event
----	-----	-----	----	-----
1	N/A	ASNL_PROTO_SIP	"Plutus" <sip:1111003@primary>	all
2	N/A	ASNL_PROTO_SIP	sip:1111003@primaryappserver1	as-feature

Client	EXPIRES(sec)	EVENT
=====	=====	=====
1111003	0	as-feature-event
Client	EXPIRES(sec)	EVENT
=====	=====	=====
1234	0	message-summary

Tips to Troubleshoot

Use the following commands to troubleshoot SUBSCRIBE-NOTIFY Passthrough:

- **debug mpa events**
- **debug mpa error**
- **debug ccsip messages**
- **debug asnl events**
- **debug asnl error**
- **debug ccsip all**

Configuration Examples for SUBSCRIBE-NOTIFY Passthrough

Example: Configuring an Event List

The following example shows how to configure an event list and add an event to the list of events that have to be passed through.

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-event-list 1
Device(config-class)# event 1 message-summary
Device(config-class)# end
```

Example: Configuring SUBSCRIBE-NOTIFY Event Passthrough Globally

The following example shows how to configure the SUBSCRIBE-NOTIFY passthrough event and link the SIP event list tag number globally.

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# pass-thru subscribe-notify-events 1
Device(conf-serv-sip)# end
```

Example: Configuring SUBSCRIBE-NOTIFY Event Passthrough under a Dial Peer

The following example shows how to configure the SUBSCRIBE-NOTIFY passthrough event and link the SIP event list tag number under a dial peer.

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 123 voip
Device(config-dial-peer)# voice-class sip pass-thru subscribe-notify-events 1
Device(config-dial-peer)# end
```