

Cisco Unified Border Element Intercluster Lookup Service

The Cisco Unified Border Element (CUBE) Intercluster Lookup Service feature enables Cisco Unified Communications Manager to establish calls using Uniform Resource Identifiers (URIs.) It provides a framework for sharing information about user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped together and associated with a cluster identifier called a route string. To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. This feature works with Cisco Unified Communication Manager Version 9.5 and later.

- Feature Information for CUBE Intercluster Lookup Service, on page 1
- Information About CUBE Intercluster Lookup Service, on page 2
- How to Configure CUBE Intercluster Lookup Service, on page 4
- Configuration Examples for CUBE Intercluster Lookup Service, on page 12

Feature Information for CUBE Intercluster Lookup Service

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 https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 1: Feature Information for CUBE Intercluster Look Up Service

Feature Name	Releases	Feature Information
CUBE Intercluster Lookup Service (ILS)	Baseline Functionality	The CUBE Intercluster Lookup Service feature enables Cisco Unified Communications Manager to establish calls using Uniform Resource Identifiers (URIs.) It provides a framework for sharing information about user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped and associated with a cluster identifier called a route string. To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. This feature works with Cisco Unified Communication Manager Version 9.5 and later. The following commands were introduced or modified: **call-route**, **destination route-string**,**passthru-hdr**, **voice class route-string**, **voice class sip-hdr-passthrulist**, **voice-class sip call-route**,**show call active voice**, **show call history voice**.

Information About CUBE Intercluster Lookup Service

CUBE Intercluster Lookup Service Overview

A Uniform Resource Identifier (URI) is a device-independent user address. A subscriber can use a URI as a personal identity and move from one network to another without any change in the URI. You cannot summarize URIs within an enterprise network (for example, abc@company.com) the same way that directory number ranges are summarized.

The Intercluster Lookup Services is a dynamic mechanism to discover URIs. When it is enabled, Cisco Unified Communications Manager users can initiate calls using URIs. The Intercluster Loookup Service provides a framework for sharing user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped together and associated with a cluster identifier called a route string. These URI groups and their associated route strings are shared between all other participating clusters.

While initiating a call, the URI uses the Intercluster Lookup Service to identify the target URI and associated route string to route the call between clusters. Cisco Unified Communications Manager uses a Session Initiation Protocol (SIP) route pattern to match the route string returned by Intercluster Lookup Service and route the call over a SIP trunk. If Intercluster Lookup Service is enabled, the Cisco Unified Communications Manager SIP trunk sends the SIP invite message with destination route string header information.

To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. CUBE supports exact match and wildcard match for a route string and parses the received destination route string header and routes a call forward to the destination. The destination can be a Cisco Unified Communications Manager cluster, public switched telephone network (PSTN), or any third-party unified communications device.

The dial-peer module is enhanced to support the dial-peer matching based on the destination route string header. The destination route string is used to match an outbound dial peer. The match can be an exact match or wildcard match.

For example, consider London.UK.EU as the route string. The SIP dial-peer configuration is as follows:

- Dial-peer 1: London.UK.EU
- Dial-peer 2: *.UK.EU
- Dial-peer 3: *.EU

The destination route string header and route string match are not case-sensitive. In this scenario, London.UK.EU and london.uk.eu match dial-peer 1 and therefore, dial-peer 1 is selected for outbound process.

If call routing policies are enabled, call routing based on a destination route string takes precedence over any other routing configurations. For example, if call routing is configured on a destination route string globally or at the dial-peer level, the call is routed considering the destination route string. If no match is found, then the call is routed using other URLs and header configuration options.

CUBE Support for URIs

For URI dialing from the Cisco Unified Communications Manager phone, use the URI in user@dest-route-string format. By default, CUBE supports only numeric E164 numbers in the user-part of the request line and headers (For example, +123456789@dest-route-string). As an administrator, you can leverage the CUBE feature Domain-Based Routing's **call-route url** command by enabling support for the alphanumeric user-part in the request line. Without this command, an alphanumeric URI fails call routing on CUBE with a 484 Address Incomplete error.

For more information on Domain-Based Routing feature, see https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/configuration/cube-book/voi-domainbased-routing.html.

Similarly, the URI-Based Dialing Enhancements feature includes support for call routing on CUBE when the user-part of the incoming request URI is non-E164. By default, the CUBE converts the @dest-route-string format of the request URI to the session target IP address of the outbound dial-peer. You can configure CUBE to pass through the full SIP URI (@dest-route-string) from the inbound call-leg without modification by using the URI-Based Dialing Enhancement's **requri-passing** command. In addition, you can use URI information to route calls using the **session target sip-uri** command.

For more information on URI-Based Dialing Enhancements feature, see https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/configuration/cube-book/sip-supp-uri-based-dialing.html.

How to Configure CUBE Intercluster Lookup Service

Configuring a Route String Pattern

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class route-string** *tag*
4. **pattern** *string*
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class route-string** *tag* **Example:** `Device(config)# voice class route-string 2`	Enters voice class configuration mode.
Step 4	**pattern** *string* **Example:** `Device(config-class)# pattern london.uk.eu`	Configures a pattern string in the specified route string. **Note** Multiple patterns can be configured under o configured under multiple dial-peers. You al provisioning the pattern.
Step 5	**end** **Example:** `Device(config-class)# end`	Exits voice class configuration mode and returns to privileged EXEC mode.

Configuring a Call Route on a Destination Route String Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **call-route dest-route-string**
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • 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 service voip`	Enters voice service configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**call-route dest-route-string** **Example:** `Device(conf-serv-sip)# call-route dest-route-string`	Configures call routing globally on a destination route string. **Note** By default, call routing on a destination
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Exits SIP configuration mode and returns to privileged EXEC mode.

Configuring a Route String Passthrough List Header

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-hdr-passthrulist** *tag*
4. **passthru-hdr** *name*
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class sip-hdr-passthrulist** *tag* **Example:** `Device(config)# voice class sip-hdr-passthrulist 2`	Enters voice class configuration mode.
Step 4	**passthru-hdr** *name* **Example:** `Device(config-class)# passthru-hdr x-cisco-dest-route-string`	Configures header to be added to the route string passthrough list.
Step 5	**end** **Example:** `Device(config-class)# end`	Exits voice class configuration mode and returns to privileged EXEC mode.

Configuring a Destination Route String Call Route at the Dial-Peer Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **description** *string*
5. **destination route-string** *tag*
6. **session protocol sipv2**
7. **session target ipv4:***destination address*
8. **voice-class sip call-route dest-route-string**
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Enters dial peer voice configuration mode.
Step 4	**description** *string* **Example:** `Device(config-dial-peer)# description outbound-dialpeer`	Adds descriptive information about the dial peer.
Step 5	**destination route-string** *tag* **Example:** `Device(config-dial-peer)# destination route-string 2`	Configures a destination route string for the dial peer. **Note** By default, the call route on a destinatio configuration at the dial-peer level takes

	Command or Action	Purpose
Step 6	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the IETF Session Initiation Protocol (SIP) for the dial peer.
Step 7	**session target ipv4:***destination address* **Example:** `Device(config-dial-peer)# session target ipv4:192.0.2.6`	Configures the session target IP address of the dial peer.
Step 8	**voice-class sip call-route dest-route-string** **Example:** `Device(config-dial-peer)# voice-class sip call-route dest-route-string`	Configures call routing on the destination route string for a dial peer.
Step 9	**end** **Example:** `Device(config-dial-peer)# end`	Exits dial peer voice configuration mode and returns to privileged EXEC mode.

Configuring a Route String Header Pass-Through Using Pass-Through List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-hdr-passthrulist** *list-tag*
4. **passthru-hdr** *header-name*
5. **passthru-hdr-unsupp**
6. **exit**
7. **dial-peer voice** *tag* **voip**
8. **description** *string*
9. **session protocol sipv2**
10. **voice-class sip pass-thru headers** *list-tag*
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class sip-hdr-passthrulist** *list-tag* **Example:** `Device(config)# voice class sip-hdr-passthrulist 101`	Configures list of headers to be passed through and enters voice class configuration mode.
Step 4	**passthru-hdr** *header-name* **Example:** `Device(config-class)# passthru-hdr Resource-Priority`	Adds header name to the list of headers to be passed through. Repeat this step for every non-mandatory header.
Step 5	**passthru-hdr-unsupp** **Example:** `Device(config-class)# passthru-hdr-unsupp`	Adds the unsupported headers to the list of headers to be passed through.
Step 6	**exit** **Example:** `Device(config-class)# exit`	Exits the current configuration session and returns to global configuration mode.
Step 7	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Enters dial peer voice configuration mode.
Step 8	**description** *string* **Example:** `Device(config-dial-peer)# description inbound-dialpeer`	Adds descriptive information about the dial peer.
Step 9	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the IETF Session Initiation Protocol (SIP) for the dial peer.
Step 10	**voice-class sip pass-thru headers** *list-tag* **Example:** `Device(config-dial-peer)# voice-class sip pass-thru headers 101`	Enables call routing based on the destination route string for a dial peer.

	Command or Action	Purpose
Step 11	**end** **Example:** `Device(config-dial-peer)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Verifying CUBE Intercluster Lookup Service Configuration

The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show voice class route-string**
3. **show call active voice**
4. **show call history voice**
5. **show sip call**

DETAILED STEPS

Procedure

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show voice class route-string**

Displays the call route-string status for voice ports.

Example:

```
Device# show voice class route-string
voice class route-string 2:
  pattern london.uk.eu
  configured in dial-peers: 7 4 6
```

Step 3 **show call active voice**

Displays call information for voice calls in progress. The sample output below shows the destination route string configuration.

Example:

```
Device# show call active voice
DestinationRouteStr=london.uk.eu
```

Step 4 **show call history voice**

Displays the call history table for voice calls. The sample output below shows the destination route string configuration.

Example:

```
Device# show call history voice | in Des
DestinationRouteStr=london.uk.eu
```

Step 5 **show sip call**

Displays active user agent client (UAC) and user agent server (UAS) information on SIP calls.

Example:

```
Device# show sip call
Total SIP call legs:2, User Agent Client:1, User Agent Server:1
SIP UAC CALL INFO
Call 1
SIP Call ID                : 5A4CAE55-E48D11E2-802BDD60-8693A1D1@192.0.2.1
   State of the call       : STATE_ACTIVE (7)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 345111
   Called Number           :
   Bit Flags               : 0xC04018 0x10000100 0x80
   CC Call ID              : 12
   Source IP Address (Sig ): 192.0.2.1
   Destn SIP Req Addr:Port : [192.0.2.6]:5060
   Destn SIP Resp Addr:Port: [192.0.2.6]:5060
   Destination Name        : 192.0.2.6
   Number of Media Streams : 1
   Number of Active Streams: 1
   RTP Fork Object         : 0x0
   Media Mode              : flow-through
   Media Stream 1
     State of the stream      : STREAM_ACTIVE
     Stream Call ID           : 12
     Stream Type              : voice-only (0)
     Stream Media Addr Type   : 1
     Negotiated Codec         : g711ulaw (160 bytes)
     Codec Payload Type       : 0
     Negotiated Dtmf-relay    : inband-voice
     Dtmf-relay Payload Type  : 0
     QoS ID                   : -1
     Local QoS Strength       : BestEffort
     Negotiated QoS Strength  : BestEffort
     Negotiated QoS Direction : None
     Local QoS Status         : None
     Media Source IP Addr:Port: [192.0.2.1]:16406
     Media Dest IP Addr:Port  : [192.0.2.6]:6020


Options-Ping    ENABLED:NO    ACTIVE:NO
   Number of SIP User Agent Client(UAC) calls: 1

SIP UAS CALL INFO
Call 1
SIP Call ID                : 1-27273@192.0.2.6
   State of the call       : STATE_ACTIVE (7)
```

```
    Substate of the call    : SUBSTATE_NONE (0)
    Calling Number          : 345111
    Called Number           : alice
    Bit Flags               : 0xC0401C 0x10000100 0x4
    CC Call ID              : 11
    Source IP Address (Sig ): 192.0.2.1
    Destn SIP Req Addr:Port : [192.0.2.6]:5061
    Destn SIP Resp Addr:Port: [192.0.2.6]:5061
    Destination Name        : 192.0.2.6
    Destination Route String: london.uk.eu  //This is the configured dest-route-string pattern.//
    Number of Media Streams : 1
    Number of Active Streams: 1
    RTP Fork Object         : 0x0
    Media Mode              : flow-through
    Media Stream 1
      State of the stream      : STREAM_ACTIVE
      Stream Call ID           : 11
      Stream Type              : voice-only (0)
      Stream Media Addr Type   : 1
      Negotiated Codec         : g711ulaw (160 bytes)
      Codec Payload Type       : 0
      Negotiated Dtmf-relay    : inband-voice
      Dtmf-relay Payload Type  : 0
      QoS ID                   : -1
      Local QoS Strength       : BestEffort
      Negotiated QoS Strength  : BestEffort
      Negotiated QoS Direction : None
      Local QoS Status         : None
      Media Source IP Addr:Port: [192.0.2.1]:16404
      Media Dest IP Addr:Port  : [192.0.2.6]:6000


Options-Ping    ENABLED:NO    ACTIVE:NO
   Number of SIP User Agent Server(UAS) calls: 1
```

Configuration Examples for CUBE Intercluster Lookup Service

Example: Configuring a Route String Pattern

```
Device> enable
Device# configure terminal
Device(config)# voice class route-string 2
Device(config-class)# pattern london.uk.eu
Device(config-class)# pattern *.uk.eu
Device(config-class)# pattern *.eu
Device(config-class)# end
```

Example: Configuring a Call Route on a Destination Route String Globally

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
```

```
Device(conf-serv-sip)# call-route dest-route-string
Device(conf-serv-sip)# end
```

Example: Configuring a Route String Passthrough List Header

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-hdr-passthrulist 2
Device(config-class)# passthru-hdr x-cisco-dest-route-string
```

Example: Configuring a Destination Route String Call Route at the Dial-Peer Level

```
Device> enable
Device# configure terminal
Device# dial-peer voice 1 voip
Device(config-dial-peer)# description outbound-dialpeer
Device(config-dial-peer)# destination route-string 2
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# session target ipv4:192.0.2.6
Device(config-dial-peer)# voice-class sip call-route dest-route-string
```

Example: Configuring a Route String Header Pass-Through Using Pass-Through List

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-hdr-passthrulist 101
Device(config-class)# passthru-hdr X-hdr-1
Device(config-class)# passthru-hdr Resource-Priority
Device(config-class)# passthru-hdr-unsupp
Device(config-class)# exit
Device(config)# dial-peer voice 1 voip
Device(config-dial-peer)# description inbound-dialpeer
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# voice-class sip pass-thru headers 101
Device(config-dial-peer)# end
```