

Cause Code Mapping

- Overview, on page 1
- Cause Code Mapping, on page 2
- Configure Cause Code Mapping, on page 3
- Verify Cause Code Mapping, on page 4

Overview

With the Cause Code Mapping feature, the NOTIFY message sent by Cisco Unified Border Element (CUBE) to a Customer Voice Portal (CVP) contains a proper reason for failure of call transfer based on the information received by CUBE from the caller instead of a 503 Service Unavailable message for all scenarios.

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 Cause Code Mapping

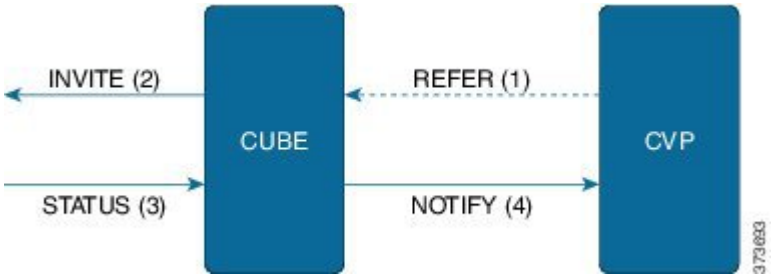
Feature Name	Releases	Feature Information
Cause Code Mapping	Cisco IOS XE 3.14S Cisco IOS 15.5(1)S3 Cisco IOS 15.5(2)S1 Cisco IOS 15.4(3)S4	With the Cause Code Mapping feature, the NOTIFY message sent by CUBE to a Customer Voice Portal (CVP) contains a proper reason for failure of call transfer based on the information received by CUBE from the caller. Following are the cause codes supported: • **17**—486 Busy Here • **19**—503 Service Unavailable • **21**—403 Forbidden • **31**—480 Temporarily Unavailable • **102**—504 Server Time-out

Cause Code Mapping

If CUBE is configured to consume REFERs that it receives, the following actions occur:

1. CUBE consumes the REFER that it receives from a Customer Voice Portal (CVP).
2. CUBE sends an INVITE (instead of a REFER) to the outbound leg (towards the caller).
3. CUBE receives a status from the caller.
4. CUBE sends a NOTIFY message to the CVP.

Figure 1: Refer Consume in CUBE



Previously, the NOTIFY message sent in step 4 included a 503 Service Unavailable message irrespective of the reason for failure of call transfer in step 3.

With the Cause Code Mapping feature, the NOTIFY message contains proper reason for failure of call transfer so that the CVP can take an appropriate action.

Table 2: Cause Code Mappings

Status Message received by CUBE (Step 3)	Cause Code	Notify message sent to CVP (Step 4)
486	17	486 Busy Here
480	31	480 Temporarily Unavailable
403	21	403 Forbidden
480	19	503 Service Unavailable
504	102	504 Server Time-out
404	1	404 Not Found
480	20	480 Temporarily Unavailable
484	28	484 Address Incomplete
502	27	502 Bad Gateway
503	38	503 Service Unavailable

Note Cause code mappings for cause code 19 and 21 require configurations mentioned in Configure Cause Code Mapping, on page 3.

Note This mapping is only for the REFER consume scenario and not for REFER passthrough.

Configure Cause Code Mapping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **reason-header override**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Device(config)# sip-ua`	Enters the SIP user agent configuration mode.
Step 4	**reason-header override** **Example:** `Device(config-sip-ua)# reason-header override`	Configures the sending of a proper reason for failure of call transfer in the NOTIFY message so that the Customer Voice Portal (CVP) can take an appropriate action.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-sip-ua)# end`	Exits to privileged EXEC mode.

Verify Cause Code Mapping

SUMMARY STEPS

1. Enter the following:
   - **debug ccsip function**
   - **debug ccsip message**
   - **debug voip application state**
   - **debug voip application core**
   - **debug voip ccapi inout**

DETAILED STEPS

Procedure

Enter the following:

- **debug ccsip function**
- **debug ccsip message**
- **debug voip application state**
- **debug voip application core**
- **debug voip ccapi inout**

Example:

486 Received by CUBE:

```
Received:
SIP/2.0 486 Busy Here
Via: SIP/2.0/UDP 9.40.3.231:5060;branch=z9hG4bK1C15625F7
From: <sip:2222@9.40.3.231>;tag=49B0964D-213C
To: <sip:3333@9.0.0.174>;tag=1
Call-ID: 7D7073E4-3F3B11E4-917BF9A9-A90B2232@9.40.3.231
CSeq: 101 INVITE
Allow-Events: telephone-event
Content-Length: 0
Reason: Q.850;cause=17
```

486 Busy here response sent in NOTIFY by CUBE

```
Sent:
NOTIFY sip:1111@9.0.0.174:9000 SIP/2.0
Via: SIP/2.0/UDP 9.40.3.231:5060;branch=z9hG4bK1C1571767
```

```
From: <sip:2222@9.40.3.231:5060>;tag=49B08E64-1374
To: <sip:1111@9.0.0.174>;tag=1
Call-ID: 1-25970@9.0.0.174
CSeq: 102 NOTIFY
Max-Forwards: 70
Date: Fri, 19 Sep 2014 13:55:46 GMT
User-Agent: Cisco-SIPGateway/IOS-15.5.20140712.124355.
Event: refer
Subscription-State: terminated;reason=noresource
Contact: <sip:2222@9.40.3.231:5060>
Content-Type: message/sipfrag
Content-Length: 25

SIP/2.0 486 Busy here
```