CHAPTER 15

Diagnostic Tests

Revised: July 2010, OL-23033-01

Introduction

This chapter describes diagnostic tests that can be performed on media gateways, subscriber terminations, and trunk terminations. All media gateways and subscriber and trunk terminations must be in the maintenance state for testing. The following tests are described in this chapter:

Caution The use of the UNIX **ifconfig down** command on any signaling interface to test or troubleshoot network or interface failures of the Cisco BTS 10200 Softswitch Signaling Interface might lead to undesirable consequences or conditions.

Media Gateway Tests

This section describes the tests that can be performed on media gateways. A gateway must be in the maintenance state.

Step 1 Force the media gateway into maintenance state:

```
control mgw id=c2421.65; mode=forced; target-state=maint;
```

Reply Example:

```
Reply: Success: CLI change successful

MGW ID -> c2421.65
INITIAL STATE -> ADMIN_INS
REQUEST STATE -> ADMIN_MAINT
RESULT STATE -> ADMIN_MAINT
FAIL REASON -> ADM found no failure
REASON -> ADM executed successful
RESULT -> ADM configure result in success
```

Step 2 Display the Test Menu.

```
diag mgw
```

Reply Example:

```
Reply: Diagnostic MGW Menu.
===
(1) MGW Network Connectivity Test
(2) MGW MGCP Connectivity Test
(3) ALL
```

Note Test 1 tests if there is a path to the device (ping).

Test 2 tests if Media Gateway Control Protocol (MGCP) has access to the device.

Test 3 performs tests 1 and 2.

Step 3 To perform a specific test, use the following examples as a guide.

```
diag mgw id=ubr-03; test=1;
```

Reply Example:

```
MEDIA GATEWAY LINE DIAGNOSTIC TEST EXECUTED -> diag mgw
ID -> ubr-03
TEST-TYPE -> ADM-MGW-NETW-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED
Reply: Diagnostic command executed.
```

```
diag mgw id=ubr-03; test=2;
```

Reply Example:

```
MEDIA GATEWAY LINE DIAGNOSTIC TEST EXECUTED -> diag mgw
ID -> ubr-03
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED
Reply: Diagnostic command executed.
```

```
diag mgw id=ubr-03; test=3;
```

Reply Example:

```
MEDIA GATEWAY LINE DIAGNOSTIC TEST EXECUTED -> diag mgw
ID -> ubr-03
TEST-TYPE -> ADM-MGW-NETW-CONNECTIVITY-TEST
TEST-DURATION -> 11
RESULT -> TEST-SUCCESS
REASON -> PASSED

MEDIA GATEWAY LINE DIAGNOSTIC TEST EXECUTED -> diag mgw
ID -> ubr-03
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED
Reply: Diagnostic command executed.
```

Step 4 Force the media gateway into INS state:

```
control mgw id=c2421.65; mode=forced; target-state=ins;
```

Reply Example:

```
Reply: Success: CLI change successful

MGW ID -> c2421.65
INITIAL STATE -> ADMIN_MAINT
REQUEST STATE -> ADMIN_INS
RESULT STATE -> ADMIN_INS
FAIL REASON -> ADM found no failure
REASON -> ADM executed successful
RESULT -> ADM configure result in success
```

Subscriber Termination Tests

This section describes the tests that can be performed on subscriber terminations. All terminations must be in the maintenance state.

Step 1 Force the subscriber termination into maintenance state:

```
control subscriber-termination id=sub2-ctx2; mode=forced; target-state=maint;
```

Step 2 Display the Test Menu.

```
diag subscriber-termination;
```

Reply Example:

```
Reply: Diagnostic Subscriber Menu.
===
(1) Subscriber MGCP Connectivity Test
(2) Subscriber Termination Connection Test
(3) Subscriber Termination Ring Test
(4) ALL
```

Note Test 1 tests if MGCP has access to the termination.

Test 2 tests if there is a path to the device (ping).

Test 3 tests if the subscriber can be rung. The Ring parameter must be specified in seconds for this test. The default is 5 seconds.

Test 4 performs tests 1 through 3.

Step 3 To perform a specific test, use the following examples as a guide.

```
diag subscriber-termination id=sub2-ctx2; test=1;
```

Reply Example:

```
SUBSCRIBER LINE DIAGNOSTIC TEST EXECUTED -> diag subscriber-termination
ID -> sub2-ctx2
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 10
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

```
diag subscriber-termination id=sub-ubr3-1@cisco.com; test=2;
```

Reply Example:

```
SUBSCRIBER LINE DIAGNOSTIC TEST EXECUTED -> diag subscriber-termination
ID -> sub-ubr3-1@Cisco.com
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 55
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

```
diag subscriber-termination id=sub-ubr3-1@cisco.com; test=3; ring-duration=10;
```

Reply Example:

```
SUBSCRIBER LINE DIAGNOSTIC TEST EXECUTED -> diag subscriber-termination
ID -> sub-ubr3-1@Cisco.com
TEST-TYPE -> ADM-TERM-RING-TEST
TEST-DURATION -> 9989
RESULT -> TEST-SUCCESS
REASON -> PASSED
Reply: Diagnostic command executed.
```

Step 4 Force the subscriber termination into INS state:

```
control subscriber-termination id=sub2-ctx2; mode=forced; target-state=ins;
```

Note Ring-duration values are 0–999 (Default = 5). Maximum ring time is 30 seconds regardless of whether the duration is set higher than or equal to 31.

Signaling System 7 Trunk Termination Tests

This section describes the tests that can be performed on Signaling System 7 (SS7) trunk terminations. All terminations must be in the maintenance state for testing.

Step 1 Force the SS7 trunk termination into maintenance state:

```
control ss7-trunk-termination tgn-id=103; mode=forced; target-state=maint;
```

Note Set customer-originated trace (COT), circuit verification message (CVM), and circuit query message (CQM) on the terminating gateway or switch to perform these tests. Otherwise, the test or tests will fail.

Step 2 Display the Test Menu.

```
diag ss7-trunk-termination
```

Reply Example:

```
Reply: Diagnostic SS7 Trunk Group Menu.
===
Test 1: SS7 MGCP Connectivity Test
Test 2: SS7 Termination Connection Test
Test 3: SS7 COT Test
Test 4: SS7 CQM Test
Test 5: SS7 CVT Test
Test 6: SS7 CIC Audit
Test 0: ALL Tests
```

Note Test 1 tests if MGCP has access to the SS7 trunk termination.

Test 2 tests if there is a path to the device (ping).

Test 3 tests the integrity of the SS7 Bearer Path.

Test 4 queries the SS7 circuit (or group of circuits) status. A range of CICs can be specified (to a maximum of 24). Both remote and local trunk states are displayed in the results.

Test 5 tests to ensure that each end of the circuit has sufficient and consistent information for using the circuit in call connections. Common language location identifier (CLLI) names are included.

Test 6 tests to ensure the CIC connections.

Test 0 performs tests 1 through 6.

Step 3 To perform a specific test, use the following examples as a guide:

```
diag ss7-trunk-termination tgn-id=103; cic=13; test=1;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 103
CIC -> 13
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

```
diag ss7-trunk-termination tgn-id=103; cic=13; test=2;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 103
CIC -> 13
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 33
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

```
diag ss7-trunk-termination tgn-id=103; cic=14; test=3;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 103
CIC -> 14
TEST-TYPE -> ADM-SS7-COT-TEST
TEST-DURATION -> 0
RESULT -> TEST-FAILURE
REASON -> ADM-MAINT-STATE-REQUIRED
Reply: Diagnostic command executed.
```

```
diag ss7-trunk-termination tgn-id=2; cic=1-24; test=4
```

Reply Example:

```
Reply: Success:

TGN ID -> 2
START CIC -> 1
END CIC -> 24
TEST TYPE -> ADM running SS7 circuit query message test
TEST DURATION -> 0
RESULT -> ADM ran test successfully
REASON -> CQM test pass
CIC COUNT -> 24
CIC STATES ->

        Remote State         Local State
        CIC 1 -> CS_IDLE     ACTV   IDLE
        CIC 2 -> CS_IDLE     ACTV   IDLE
        CIC 3 -> CS_IDLE     ACTV   IDLE
        CIC 4 -> CS_IDLE     ACTV   IDLE
        CIC 5 -> CS_IDLE     ACTV   IDLE
        CIC 6 -> CS_IDLE     ACTV   IDLE
        CIC 7 -> CS_IDLE     ACTV   IDLE
        CIC 8 -> CS_IDLE     ACTV   IDLE
        CIC 9 -> CS_IDLE     ACTV   IDLE
        CIC 10 -> CS_IDLE    ACTV   IDLE
        CIC 11 -> CS_IDLE    ACTV   IDLE
        CIC 12 -> CS_IDLE    ACTV   IDLE
        CIC 13 -> CS_IDLE    ACTV   IDLE
        CIC 14 -> CS_IDLE    ACTV   IDLE
        CIC 15 -> CS_IDLE    ACTV   IDLE
        CIC 16 -> CS_IDLE    ACTV   IDLE
        CIC 17 -> CS_IDLE    ACTV   IDLE
        CIC 18 -> CS_IDLE    ACTV   IDLE
        CIC 19 -> CS_IDLE    ACTV   IDLE
        CIC 20 -> CS_IDLE    ACTV   IDLE
        CIC 21 -> CS_IDLE    ACTV   IDLE
        CIC 22 -> CS_IDLE    ACTV   IDLE
        CIC 23 -> CS_IDLE    ACTV   IDLE
        CIC 24 -> CS_IDLE    ACTV   IDLE
```

Note Table 15-1 lists the responses that can be returned for the CQM test.

```
diag ss7-trunk-termination tgn-id=2; cic=1; test=5
```

Reply Example:

```
Reply: Success:

TGN ID -> 2
START CIC -> 1
END CIC -> 1
TEST TYPE -> ADM running SS7 circuit validation test
TEST DURATION -> 0
RESULT -> ADM ran test successfully
REASON -> CVT test pass
CLLI -> DALLTXRCDN5
```

Step 4 Force the SS7 trunk termination into INS state:

```
control ss7-trunk-termination tgn-id=103; mode=forced; target-state=ins;
```

Table 15-1 CQM Responses

Response	Description
CS_TRANSIENT	Transient
CS_UNEQUIPPED	Unequipped
CS_IC_BUSY	Incoming Busy
CS_IC_BUSY_LOCBLOC	Incoming Busy and Locally Maintenance Blocked
CS_IC_BUSY_REMBLOC	Incoming Busy and Remotely Maintenance Blocked
CS_IC_BUSY_BOTH_BLOC	Incoming Busy and Remotely and Locally Maintenance Blocked
CS_OG_BUSY	Outgoing Busy
CS_OG_BUSY_LOCBLOC	Outgoing Busy and Locally Maintenance Blocked
CS_OG_BUSY_REMBLOC	Outgoing Busy and Remotely Maintenance Blocked
CS_OG_BUSY_BOTH_BLOC	Outgoing Busy and Remotely and Locally Maintenance Blocked
CS_IDLE	Idle
CS_IDLE_LOCBLOC	Idle and Locally Maintenance Blocked
CS_IDLE_REMBLOC	Idle and remotely maintenance blocked
CS_IDLE_BOTH_BLOC	Idle and Remotely and Locally Maintenance Blocked
CS_HW_LOCBLOC	Locally Hardware Blocked
CS_HW_LOCBLOC_LOCBLOC	Locally Hardware and Locally Maintenance Blocked
CS_HW_LOCBLOC_REMBLOC	Locally Hardware and Remotely Maintenance Blocked
CS_HW_LOCBLOC_BOTHBLOC	Locally Hardware and Remotely and Locally Maintenance Blocked
CS_HW_REMBLOC	Remotely Hardware Blocked
CS_HW_REMBLOC_LOCBLOC	Remotely Hardware and Locally Maintenance Blocked
CS_HW_REMBLOC_REMBLOC	Remotely Hardware and Remotely Maintenance Blocked
CS_HW_REMBLOC_BOTHBLOC	Remotely Hardware and Remotely and Locally Maintenance Blocked
CS_HW_BOTHBLOC	Remotely and Locally Hardware Blocked
CS_HW_BOTHBLOC_LOCBLOC	Remotely and Locally Hardware and Locally Maintenance Blocked
CS_HW_BOTHBLOC_REMBLOC	Remotely and Locally Hardware and Remotely Maintenance Blocked
CS_HW_BOTHBLOC_BOTHBLOC	Remotely and Locally Hardware and Remotely and Locally Maintenance Blocked

Integrated Services Digital Network Trunk Termination Tests

This section describes the tests that can be performed on Integrated Services Digital Network (ISDN) trunk terminations. All terminations must be in the maintenance state for testing.

Step 1 Force the ISDN trunk termination into maintenance state:

```
control isdn-trunk-termination tgn-id=17; mode=forced; target-state=maint;
```

Step 2 Display the Test Menu.

```
diag isdn-trunk-termination
```

Reply Example:

```
Reply: Diagnostic ISDN Trunk Group Menu.
===
(1) ISDN MGCP Connectivity Test
(2) ISDN Termination Connection Test
(3) ALL
```

Note Test 1 tests if MGCP has access to the ISDN termination.

Test 2 tests if there is a path to the device (ping).

Test 3 performs tests 1 and 2.

Step 3 To perform a specific test, use the following examples as a guide:

```
diag isdn-trunk-termination test=1; tgn-id=17; cic=1;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 17
CIC -> 1
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

```
diag isdn-trunk-termination test=2; tgn-id=17; cic=1;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 17
CIC -> 1
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

Step 4 Force the ISDN trunk termination into MAINT state:

```
control isdn-trunk-termination tgn-id=17; mode=forced; target-state=maint;
```

Channel-Associated Signaling Trunk Termination Tests

This section describes the tests that can be performed on channel-associated signaling (CAS) trunk terminations. All terminations must be in the maintenance state for testing.

Step 1 Force the CAS trunk termination into maintenance state:

```
control cas-trunk-termination tgn-id=64; mode=forced; target-state=maint;
```

Step 2 Display the Test Menu.

```
diag cas-trunk-termination
```

Reply Example:

```
Reply: Diagnostic CAS Trunk Group Menu.
===
(1) CAS MGCP Connectivity Test
(2) CAS Termination Connection Test
(3) ALL
```

Note Test 1 tests if MGCP has access to the CAS termination.

Test 2 tests if there is a path to the device (ping).

Test 3 performs tests 1 and 2.

Step 3 To perform a specific test, use the following examples as a guide:

```
diag cas-trunk-termination tgn-id=64; cic=1; test=1;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 64
CIC -> 1
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

```
diag cas-trunk-termination tgn-id=64; cic=1; test=2;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 64
CIC -> 1
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 32
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

```
diag cas-trunk-termination tgn-id=64; cic=1; test=3;
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 64
CIC -> 1
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 11
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510

TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 64
CIC -> 1
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 32
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

Step 4 Force the CAS trunk termination into INS state:

```
control cas-trunk-termination tgn-id=64; mode=forced; target-state=ins;
```

Announcement Trunk Termination Tests

This section describes the tests that can be performed on Announcement trunk terminations. All terminations must be in the maintenance state for testing.

Step 1 Force the Announcement trunk termination into maintenance state:

```
control annc-trunk-termination tgn-id=13; mode=forced; target-state=maint;
```

Step 2 Display the Test Menu.

```
diag annc-trunk-termination;
```

Reply Example:

```
Reply: Diagnostic ANC Trunk Group Menu.
===
(1) ANC MGCP Connectivity Test
(2) ANC Termination Connection Test
(3) ALL
```

Note Test 1 tests if MGCP has access to the announcements module (ANC) termination.

Test 2 tests if there is a path to the device (ping).

Test 3 performs tests 1 and 2.

Step 3 To perform a specific test, use the following examples as a guide:

```
diag annc-trunk-termination test=1; tgn-id=13; cic=1
```

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 13
CIC -> 1
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 0
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510
Reply: Diagnostic command executed.
```

diag annc-trunk-termination test=2; tgn-id=13; cic=1

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 13
CIC -> 1
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 33
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

diag annc-trunk-termination test=3; tgn-id=13; cic=1

Reply Example:

```
TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 13
CIC -> 1
TEST-TYPE -> ADM-MGW-MGCP-CONNECTIVITY-TEST
TEST-DURATION -> 11
RESULT -> TEST-SUCCESS
REASON -> PASSED: Reason: AUEP-NACK received with RespCode = 510

TRUNK DIAGNOSTIC TEST EXECUTED -> diag trunk
TG-NUM -> 13
CIC -> 1
TEST-TYPE -> ADM-TERM-CONNECTION-TEST
TEST-DURATION -> 33
RESULT -> TEST-SUCCESS
REASON -> PASS successfully.
Reply: Diagnostic command executed.
```

Step 4 Force the Announcement trunk termination into INS state:

control annc-trunk-termination tgn-id=13; mode=forced; target-state=ins;

Troubleshooting Using Snoop

Caution Snoop should not be used on the Cisco BTS 10200 call agent itself in a production network. It can cause performance degradation.

Snoop can be used on the Cisco BTS 10200 call agent during test and turn-up phase during very low call volume periods. Snoop can always be used on a separate UNIX machine connected to a switch that has been properly set up for port span/mirroring. You must be logged in as "root" user to run snoop. Snoop can be used to decode text protocols or can be saved to a file and opened with Ethereal when binary protocols are used. Ethereal is open source software and can be downloaded from http://www.ethereal.com. To use Snoop to diagnose network problems, take the following steps:

Step 1 Find all routes to the destination in question. There are probably multiple roots, so multiple interfaces will need to be snooped. (Skip this step if you are snooping from a separate UNIX machine–you will just snoop the span destination interface in that case.) In this example, destination Internet Protocol (IP) 10.0.0.1 is in question. The fully qualified domain name (FQDN) can be used if it is resolvable by domain name system (DNS). Issue the **snoop** command several times as there may be redundant routes.

```
mssol-ca0-a# route get 10.0.0.1
   route to: 10.0.0.1
destination: default
       mask: default
    gateway: 10.0.0.253
 interface: qfe4
      flags: <UP,GATEWAY,DONE>
 recvpipe sendpipe ssthresh    rtt,ms rttvar,ms hopcount      mtu     expire
       0         0         0         0         0         0      1500         0
mssol-ca0-a# route get 10.0.0.1
   route to: 10.0.0.1
destination: default
       mask: default
    gateway: 10.0.0.253
 interface: qfe4
      flags: <UP,GATEWAY,DONE>
 recvpipe sendpipe ssthresh    rtt,ms rttvar,ms hopcount      mtu     expire
       0         0         0         0         0         0      1500         0
mssol-ca0-a# route get 10.0.0.1
   route to: 10.0.0.1
destination: default
       mask: default
    gateway: 10.20.0.253
 interface: qfe0
      flags: <UP,GATEWAY,DONE>
 recvpipe sendpipe ssthresh    rtt,ms rttvar,ms hopcount      mtu     expire
       0         0         0         0         0         0      1500         0
mssol-ca0-a# route get 10.0.0.1
   route to: 10.0.0.1
destination: default
       mask: default
    gateway: 10.20.0.253
 interface: qfe0
      flags: <UP,GATEWAY,DONE>
 recvpipe sendpipe ssthresh    rtt,ms rttvar,ms hopcount      mtu     expire
       0         0         0         0         0         0      1500         0
```

Note Each interface reported above must be snooped to catch all packets across redundant routes. In the example, interfaces qfe0 and qfe4 must be snooped.

Step 2 Issue the **snoop** command. The syntax might differ depending on protocol(s) that are being analyzed.

Session Initiation Protocol (SIP) example:

10.0.0.1 is a SIP phone. The goal is to monitor the SIP traffic between the Cisco BTS 10200 and the SIP phone.

```
# snoop -d qfe0 -x 42 host 10.0.0.1 and port 5060 and udp &
# snoop -d qfe4 -x 42 host 10.0.0.1 and port 5060 and udp &
```

MGCP/network-based call signaling (NCS) example:

10.0.0.1 is an integrated access device (IAD) running MGCP. The goal is to monitor MGCP traffic between the Cisco BTS 10200 and the IAD.

```
# snoop -d qfe0 -x 42 host 10.0.0.1 and port 2427 and udp &
# snoop -d qfe4 -x 42 host 10.0.0.1 and port 2427 and udp &
```

Stream Control Transmission Protocol (SCTP)/MTP3 user adaptation (M3UA)/ISDN user part (ISUP) example:

Since these protocols are not text based like those mentioned above, use the -o option with snoop to capture packets in an Ethereal readable format. Ethereal can decode SCTP/M3UA/ISUP or SCTP/SCCP user adapter (SUA)/Transaction Capabilities Application Part (TCAP). 10.0.0.1 is a Signaling Gateway acting as an M3UA peer with the Cisco BTS 10200.

```
# snoop -d qfe0 -o sctp.cap host 10.0.0.1 (this will capture all traffic)
```

Step 3 Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect. To capture sctp packets that contain M3UA information:

a. First, find the port M3UA will use to communicate with the signaling gateway (SG).

```
CLI> show sctp-assoc platform-id=CA146

ID=sgp1-itpa
SGP_ID=sgp1
SCTP_ASSOC_PROFILE_ID=sctp-prof1
REMOTE_PORT=2905 <--------------------this port
REMOTE_TSAP_ADDR1=10.0.0.1
PLATFORM_ID=CA146
DSCP=NONE
IP_TOS_PRECEDENCE=CRITICAL
LOCAL_RCVWIN=3000
MAX_INIT_RETRANS=3
MAX_INIT_RTO=500
STATUS=INS
ULP=XUA

# snoop -d qfe0 -o m3ua.cap host 10.0.0.1 and port 2905
```

b. Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect.

SCTP/SUA/TCAP example 1:

10.0.0.1 is a Signaling Gateway acting as an SUA peer with the Cisco BTS 10200. The goal is to capture all 800/local number portability (LNP) queries.

a. Follow the same syntax as for the M3UA case, except find which port SUA uses to communicate with the SG for Advanced Intelligent Network (AIN) features:

```
CLI> show sctp-assoc platform-id=FSAIN205

ID=sctp-ain-itpa
SGP_ID=sgp1
SCTP_ASSOC_PROFILE_ID=sctp-prof1
REMOTE_PORT=2907 <-------------------this port
REMOTE_TSAP_ADDR1=10.0.0.1
PLATFORM_ID=FSAIN205
DSCP=NONE
IP_TOS_PRECEDENCE=CRITICAL
LOCAL_RCVWIN=3000
MAX_INIT_RETRANS=3
MAX_INIT_RTO=500
STATUS=INS
ULP=XUA

# snoop -d qfe0 -o suaain.cap host 10.0.0.1 and port 2907
```

b. Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect.

SCTP/SUA/TCAP example 2:

10.0.0.1 is a Signaling Gateway acting as an SUA peer with the Cisco BTS 10200. The goal is to capture all offnet automatic callback and automatic rollback (ACAR) queries.

a. Follow the same syntax as for the M3UA case, except find the port SUA uses to communicate with the SG for plain old telephone service (POTS) features:

```
CLI> show sctp-assoc platform-id=FSPTC235

ID=sctp-ptc-itpa
SGP_ID=sgp2
SCTP_ASSOC_PROFILE_ID=sctp-prof1
REMOTE_PORT=2906 <-------------------this port
REMOTE_TSAP_ADDR1=10.0.0.1
PLATFORM_ID=FSPTC235
DSCP=NONE
IP_TOS_PRECEDENCE=FLASH
LOCAL_RCVWIN=64000
MAX_INIT_RETRANS=5
MAX_INIT_RTO=1000
STATUS=INS
ULP=XUA

# snoop -d qfe0 -o suapots.cap host 10.0.0.1 and port 2906
```

b. Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect.

H.323 Protocol (H323) example:

10.0.0.1 is an H323 gateway. 10.0.0.129 is an H323 gatekeeper. Our goal is to monitor Registration, Admissions, Status (RAS), and H.225 messaging.

a. First, find the RAS port number and the H.225 port number.

```
CLI> show h323-gw

ID=ccm3_gw1
STATUS=INS
OPER_STATUS=NF
GW_H225_PORT=1720 <----------- this port
TGN_ID=4441
SECURITY=N
H245_TUNNELING=DEFAULT
TCP_MAX_LIMIT=5
TCP_MAX_AGE=30
MAX_VOIP_CALLS=65535
HIGH_WATER_MARK=0
LOW_WATER_MARK=0
IRR_BANDWIDTH_SUPP=N
IPTOS_SIG_LOWDELAY=Y
IPTOS_SIG_THROUGHPUT=N
IPTOS_SIG_RELIABILITY=N
IPTOS_SIG_PRECEDENCE=FLASH
BRQ_SUPP=Y
ANNEXE_RETRANSMIT_TIMER=500
ANNEXE_RETRANSMIT_MULTIPLIER=2
ANNEXE_RETRANSMIT_ATTEMPTS=8
CALL_START_MODE=FAST_START
ANNEXE_SUPP=N
ANNEXR_SUPP=N
STATUS_ENQ_TIMER=4
CODEC_NEG_TIMER=200
CODEC_NEG_ATTEMPTS=4
SOURCE_BASED_ROUTING=NONE

CLI> show h323-gw2gk

H323_GW_ID=ccm3_gw1
GK_ID=Metro-GK
PRIORITY=0
GK_IP_ADDR=10.0.0.129
GK_RAS_PORT=1719 <------------ this port
MULTICAST=N
TIME_TO_LIVE=60

# snoop -d qfe0 -o h323.cap host 10.0.0.1 and port 1720 or host 10.0.0.129 and port
1719
```

b. Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect.

COPs example:

10.0.0.1 is a cable modem termination system (CMTS) and is configured as an aggregation identification (AGGR-ID) in the Cisco BTS 10200. The goal is to monitor all Common Open Policy Service Protocol (COPS) messaging to and from the CMTS.

a. Issue the following command:

```
# snoop -d qfe0 -o cops.cap host 10.0.0.1 and port 2126 and tcp
```

b. Use **Control-C** to stop the packet capture. Open the file in Ethereal and inspect.

Step 4 Packets can be redirected to a file (not readable by Ethereal) in the following way:

```
# snoop -d qfe0 -x 42 host 10.0.0.1 and port 2427 and udp > mycapt.cap
```

Step 5 Stop the snoop processes.

```
# pkill snoop
# pgrep snoop (should not report any process ids)
```

Query Verification Tool and Translation Verification Tool

This section describes the Query Verification Tool (QVT) and the Translation Verification Tool (TVT) and is organized into the following sub-sections:

Tool Requirements

The following requirements are supported in the QVT and TVT:

- TVT—Provide a tool to find, diagnose, and trace call flow path decisions.
- Query Local Routing Number (QLRN) Tool—Provide the ability to enter a ten digit directory number and launch a query to the service control point (SCP) as though it was a number called from the signal switching point (SSP).
- Query Tool E800VER Command—Send a database query to the SCP as if it were an 800 called number from the SSP without initiating a call.
- Query Tool CNAMDVER and TESTSS CNAMD Commands—Provide the ability to query the SCP database for the calling name delivery (CNAM) display and privacy status associated with the name without initiating a call.

Query Verification Tool

This section describes the QVT and includes the following sections:

Overview

The QVT enables a user to generate TCAP queries to external databases through the command line interface (CLI) interface. The types of queries supported are:

- Line information database (LIDB)—Generated by the POTS Feature Server
- Toll-Free—Generated by the AIN Feature Server
- LNP—Generated by the AIN Feature Server

Command Format

The **QVT** command uses the following format:

```
query <lidb|toll-free|lnp> parameter=value;
```

Response Format

The system response to a query is in the following format:

```
Reply: <success|failure>; parameter=value;
```

Common Response Parameters

Successful response parameters include the following:

- OPC—Originating Point Code
- SSN—Subsystem Number
- TT—translation type
- SCP-Point-Code—Point Code of the SCP
- Automatic call gapping (ACG) component received
- ACG-Control-Code-Length
- Generic address parameter (GAP)–duration
- GAP-Interval
- Announcement-Cause-Code

An error message will be displayed if the query is not successful. For more information about error messages and problem resolution, refer to the "Query Errors" section on page 15-19.

Query Line Information Database Response Parameters

Additional parameters returned in response to a **query lidb** command include:

- Calling-DN
- Caller-ID Name String
- Caller-ID Name Privacy

Query Toll-Free Parameters

The following additional parameters are returned in response to a **query toll-free** command:

- Message-Type
- Original Number
- Translated Number
- Carrier
- Send-Notification-Received

Query Local Number Portability Parameters

The following additional parameters are returned in response to a **query LNP** command:

- Original Number
- Translated Number

Query Errors

An error can occur when a **query** command fails. This section specifies error responses and possible resolutions for problems.

Request Timeout

A query is sent to the feature server, but no response is received. The error response is similar to the one in the following example:

```
CLI> query lidb calling-dn=123247238723; opc-id=opc;
QUERY ON FEATURE SERVER FSPTC235 IS...->
FSPTC235 -> No Reply received!
Reply: Failure:
CLI>
```

The Feature Server did not respond to the query before a timeout occurred. Take the following steps to resolve the problem:

- If it was an LIDB query, execute the **nodestat** command on the POTS Feature Server to confirm that it is Active.
- If it was a Toll-Free or LNP query, execute the **nodestat** command on the AIN Feature Server to confirm that it is Active.
- If the platform is Active, execute the following command to confirm that the selective call acceptance (SCA) process is running:

  ```
  ps -aef | grep sca
  ```

If the process is not running, start it through process debug manager (PDM) or by stopping and restarting the platform.

- If the platform is Active, execute the following command to confirm that the TCAP signaling adapter (TSA) process is running:

  ```
  ps -aef | grep tsa
  ```

If the process is not running, start it through PDM or by stopping and restarting the platform.

- If the SCA and TSA processes are running on the Active platform, check the trace files for errors associated with the query.

Service Control Point Timeout

The SCP does not respond to a query. The error response is similar to the following example:

```
CLI> query lidb calling-dn=1232472387283; opc-id=opc;
QUERY ON FEATURE SERVER FSPTC235 IS...->
RESULT ->
QVT query has timed out
QUERYSTATUS -> Miscellaneous Failure
Reply: Success:
CLI>
```

There is no response from the SCP. Contact the SCP provider to find out why there is no error response returned from the SCP.

Missing Mandatory Parameter

The user performs a query but does not provide all required parameters. The error response is similar to the following example:

```
CLI> query toll-free called-dn=8002550002; user-type=calling-dn; user-id=2182640018;
lata=100; bearer-capability=speech; trigger-criteria=9;
Required attributes missing:
opc_id
CLI>
```

Supply all required parameters for the query. To view a list of parameters required for a command, enter a question mark (?) after the partial command. For example, query lidb? will display a list of required parameters for a LIDB query.

Advanced Intelligent Network 0.1 Query Attempted for IN/1 Configuration

An AIN0.1 Toll-Free query has been performed, but the system specifies that the Toll-Free subsystem is IN/1. The error response is similar to the following example:

```
CLI> query toll-free called-dn=8002550002; user-type=calling-dn; user-id=2182640018;
lata=100; bearer-capability=speech; trigger-criteria=9, opc-id=opc;
Reply: Failure: Missing CALLING_DN for the IN/1 query
CLI>
```

Reissue the command in the IN/1 format. To see what message type is specified for the Toll-Free subsystem, enter the following command:

```
CLI> query toll-free-msg-type opc-id=opc;
MESSAGE-TYPE=IN1
Reply: Success:
```

IN/1 Query Attempted for Advanced Intelligent Network 0.1 Configuration

An IN/1 Toll-Free query has been performed, but the system specifies that the Toll-Free subsystem is AIN 0.1. The error response is similar to the following example:

```
CLI> query toll-free: called-dn=8002550002; calling-dn=2182640018; lata=100;
trigger-criteria=9; opc-id=opc;
Reply: Failure: Missing USER_TYPE for the AIN 0.1 query
CLI>
```

Reissue the command in the AIN 0.1 format. To see what message type is specified for the Toll-Free subsystem, enter the following command:

```
CLI> query toll-free-msg-type; opc-id=opc;
MESSAGE-TYPE=AIN01
Reply: Success:
CLI>
```

Parameter Boundary Error

The query can fail if you enter invalid data for a specific parameter. In the following example, a value outside the boundary of expected values for the trigger-criteria parameter has been specified:

```
CLI> query toll-free; called-dn=8002550002; calling-dn=2182640018; lata=100;
trigger-criteria=12; opc-id=opc;
Invalid parameter value.
trigger_criteria=12; Enter one of the following values: [3,6,7,8,9,10].
CLI>
```

To resolve this error, enter a valid value for the specified parameter.

Record Does Not Exist

In the following example, a value has been entered for a lata that has not been provisioned:

```
CLI> query toll-free; called-dn=8002550002; calling-dn=2182640018; lata=101;
trigger-criteria=9; opc-id=opc;
Reply: Failure: LATA 101 does not exist
CLI>
```

To resolve this error, enter a provisioned local access and transport area (LATA).

Local Network Failure

When communication is lost between the Cisco BTS 10200 and the IP transfer point (ITP) gateway, a local network failure might occur. The most likely reason for this error is that the SCTP association is Out Of Service. The error response is similar to the following example:

```
CLI> query toll-free; called-dn=8002550002; calling-dn=2182640018; lata=100;
trigger-criteria=9; opc-id=opc;
QUERY ON FEATURE SERVER FSAIN205 IS...->
RESULT->
MTP failure - occurs at SP (PC7-44-1, SSN=254)
QUERYSTATUS -> Network Failure
Reply: Success:
CLI>
```

Perform the following to diagnose the problem:

- Execute the query again with the table-info option set to yes.
- Determine the status of the SCTP associations used for this command. If the associations are Out Of Service, control the associations back into service.

Remote Network Failure

A failure has occurred at a point code other than the OPC. The query response will specify what problem has occurred and at which point code the problem is detected. In the following example, the point code of the signal transfer point (STP) is reporting a failure because there is no Global Title Translation entry in the STP global title translation (GTT) database for the calling-dn.

```
CLI> query lidb; calling-dn=9823456789; opc-id=opc;
QUERY ON FEATURE SERVER FSPTC235 IS...->
RESULT ->
No translation for this specific address - occurs at SP (PC=1-101-0, SSN=0)
QUERYSTATUS -> Network Failure
Reply: Success:
CLI> status sctp-assoc;
```

To resolve this error, add an entry to the STP GTT database to translate the calling-dn and route the query request to the LIDB subsystem on the SCP.

Query Verification Tool Measurements

Table 15-2 identifies the measurements generated by the AIN Feature Server for the QVT feature.

Table 15-2 QVT AIN Tool Counters

Counter Label	Counter Description
TOOLS_LNP_QUERY_ATTMP	The total number of times the reporting feature server received a request to perform an LNP query from the QVT tool
TOOLS_LNP_QUERY_SUCC	The total number of times the reporting feature server received a request to perform an LNP query from the QVT tool and completed it successfully
TOOLS_TOLLFREE_QUERY_ATTMP	The total number of times the reporting feature server received a request to perform a Toll Free query from the QVT tool
TOOLS_TOLLFREE_QUERY_SUCC	The total number of times the reporting feature server received a request to perform a Toll Free query from the QVT tool and completed it successfully

Table 15-3 identifies the measurements generated by the POTS Feature Server for the QVT feature.

Table 15-3 QVT POTS Tool Counters

Counter Label	Counter Description
TOOLS_LIDB_QUERY_ATTMP	The total number of times the reporting feature server received a request to perform an LIDB query from the QVT tool
TOOLS_LIDB_QUERY_SUCC	The total number of times the reporting feature server received a request to perform an LIDB query from the QVT tool and completed it successfully

Translation Verification Tool

This section describes the TVT and includes the following sections:

Overview

The TVT is a diagnostic tool that simulates a call from the originator to a specific destination based on dialed digits. It enables a user to check system translations and determine if routing will occur as expected without making a call.

Command Format

The **TVT** command uses the following format:

```
translate <line|trunk>; parameter=value;
```

Response Format

Translation is the process of determining the destination of a call based on the dialed digits. The TVT performs translations and returns the tables traversed in order to reach the destination number. It does not complete a call but only allows you to view the route of the call.

The following example illustrates an incoming line call terminating to a trunk:

```
CLI> translate line calling-dn=9722331286; called-dn=7034321234;

TABLE: SUBSCRIBER

ID=sub1_ata2; CATEGORY=INDIVIDUAL; NAME=sub1; STATUS=ACTIVE; DN1=9722331003; PRIVACY=NONE;
RING_ TYPE_DN1=1; TERM_ID=a00/1; MGW_ID=ata2; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=northtexas; TERM_TYPE=TERM; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; SEND_BILLING_DN=N; SEND_BDN_AS_CPN=N; SEND_BDN_FOR_EMG=N;

TABLE: SUBSCRIBER_PROFILE

ID=northtexas; DIAL_PLAN_ID=dp1; LOCAL_PFX1_OPT=NR; TOLL_PFX1_OPT=RQ; POP_ID=1; OLI=0;
EA_USE_PIC1=Y;

TABLE: DIAL_PLAN_PROFILE

ID=dp1; Description=dialingplanprofile; NANP_DIAL_PLAN=Y; DNIS_DIGMAN_ID=dp1;

TABLE: DIAL_PLAN

ID=dp1; DIGIT_STRING=408555; DEST_ID=ssp1dest; SPLIT_NPA=NONE; DEL_DIGITS=0;
MIN_DIGITS=10; MAX_DIGITS=10; NOA=NATIONAL;
```

```
TABLE: DESTINATION
DEST_ID=ssp1dest; CALL_TYPE=LOCAL; ROUTE_TYPE=ROUTE; ROUTE_GUIDE_ID=ssp1rg; ZERO_PLUS=N;
INTRA_STATE=Y; GAP_ROUTING=N; CLDPTY_CTRL_REL_ALWD=N; TABLE: ROUTE_GUIDE ID=ssp1rg;
POLICY_TYPE=ROUTE; POLICY_ID=ssp1route;

TABLE: ROUTE
ID=ssp1route; TGN1_ID=1; DEL_DIGITS1=0; DEL_DIGITS2=0; EL_DIGITS3=0; DEL_DIGITS4=0;
DEL_DIGITS5=0; DEL_DIGITS6=0; DEL_DIGITS7=0; DEL_DIGITS8=0; DEL_DIGITS9=0; DEL_DIGITS10=0;
TG_SELECTION=RR;

TABLE: TRUNK_GRP
ID=1; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=1-12-1;
TG_PROFILE_ID=ssp1-tg-prof; STATUS=INS; DIRECTION=BOTH; SEL_POLICY=ASC; GLARE=EVEN;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=N; POP_ID=1; REMOTE_SWITCH_LRN=2122129999;
DIAL_PLAN_ID=dp19; Description=TG to BTS12; DEL_DIGITS=0; OPER_STATUS=NF;
TRAFFIC_TYPE=TANDEM; ANI_BASED_ROUTING=N; CLLI=DAL177DS3;
CALL_CTRL_ROUTE_ID=bts12-ccroute1; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;

Reply: Success:

CLI>
```

Translation Verification Tool Measurements

Table 15-4 identifies the measurements generated by the TVT Tool.

Table 15-4 TVT Tool Counters

Counter Label	Counter Description
TOOLS_LNP_QUERY_ATTMP	The total number of times the reporting feature server received a request to perform an LNP query from the QVT tool
TOOLS_LNP_QUERY_SUCC	The total number of times the reporting feature server received a request to perform an LNP query from the QVT tool and completed it successfully
TOOLS_TOLLFREE_QUERY_ATTMP	The total number of times the reporting feature server received a request to perform a toll free query from the QVT tool
TOOLS_TOLLFREE_QUERY_SUCC	The total number of times the reporting feature server received a request to perform a toll free query from the QVT tool and completed it successfully

Using Query Verification Tool and Translation Verification Tool Together

It may be necessary to use both QVT and TVT queries to diagnose routing of a call. If the results of a **translate** command indicate that a toll-free or LNP query is generated, execute the QVT query. Use the results of the QVT query to generate another TVT query.

The following example illustrates verifying routing of a call from (972) 233-1286 to (800) 255-3002:

Step 1 Execute a **TVT translate** command:

```
CLI> translate line calling-dn=9722331286; called-dn=8002553002;

TRANSLATE LINE ON CALL AGENT CA146 IS...->
TABLEINFO ->
******TOLL FREE CALL NEEDS AN 800 QUERY******

Reply: Success:

CLI>
```

Step 2 The **translate** command indicates that a Toll-Free query is required. Perform the QVT query to do the number translation.

```
CLI> query toll-free called-dn=8002553002; calling-dn=9722331286; lata=100; opc-id=opc;

TOLL FREE QUERY ON FEATURE SERVER FSAIN520 IS...->
RESULT->
OPC=7-2-1
SSN=254
TT-254
SCP-Point-Code=1-101-0
Message-Type=IN/1
Called Number=8002553002
Translated Number=7034323002
Carrier=0000

Reply: Success:

CLI>
```

Step 3 The translated number returned by the QVT query can now be used in a TVT **translate** command to verify call routing.

```
CLI> translate line calling-dn=9722331286; called-dn=7034323002;

TRANSLATE LINE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sub_1_6; CATEGORY=INDIVIDUAL; NAME=sub16; STATUS=ACTIVE; ADDRESS1=1651 n glenville
suite 200; ADDRESS2=Richardson tx 75081; BILLING_DN=9722331286; DN1=9722331286;
PRIVACY=NONE; RING_TYPE_DN1=1; TERM_ID=aaln/S1/6; MGW_ID=c2421_1; PIC1=NONE; PIC2=NONE;
PIC3=NONE; GRP=N; USAGE_SENS=Y; SUB_PROFILE_ID=sub_pmlhg_prof1; TERM_TYPE=TERM;
IMMEDIATE_RELEASE=N; TERMINATING_IMMEDIATE_REL=N; SEND_BILLING_DN=N; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N;

TABLE: SUBSCRIBER_PROFILE

ID=sub_pmlhg_prof1; DIAL_PLAN_ID=dp1; LOCAL_PFX1_OPT=NR; TOLL_PFX1_OPT=RQ; LSA=9;
POP_ID=1; OLI=0; EA_USE_PIC1=N;

TABLE: DIAL_PLAN_PROFILE
ID=dp1; Description=dialing plan profile ID 1; NANP_DIAL_PLAN=Y; DNIS_DIGMAN_ID=dp_svc;

TABLE: DIAL_PLAN

ID=dp1; DIGIT_STRING=703432; DEST_ID=ssp1-dest; SPLIT_NPA=NONE; DEL_DIGITS=0;
MIN_DIGITS=7; MAX_DIGITS=10; NOA=NATIONAL;
```

```
TABLE: DESTINATION

DEST_ID=ssp1-dest; CALL_TYPE=LOCAL; ROUTE_TYPE=ROUTE; ROUTE_GUIDE_ID=ssp1-rg; ZERO_PLUS=N;
INTRA_STATE=Y; GAP_ROUTING=N; CLDPTY_CTRL_REL_ALWD=N;

TABLE: ROUTE_GUIDE

ID=ssp1-rg; POLICY_TYPE=ROUTE; POLICY_ID=ssp1-route;

TABLE: ROUTE

ID=ssp1-route; TGN1_ID=3; DEL_DIGITS1=0; DEL_DIGITS2=0; DEL_DIGITS3=0; DEL_DIGITS4=0;
DEL_DIGITS5=0; DEL_DIGITS6=0; DEL_DIGITS7=0; DEL_DIGITS8=0; DEL_DIGITS9=0; DEL_DIGITS10=0;
TG_SELECTION=RR;

TABLE: TRUNK_GRP

ID=3; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=1-12-1;
TG_PROFILE_ID=ssp1-tg-prof; STATUS=INS; DIRECTION=BOTH; SEL_POLICY=ASC; GLARE=EVEN;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=N; POP_ID=1; REMOTE_SWITCH_LRN=2122129999;
DIAL_PLAN_ID=dp19; Description=TG to BTS12; DEL_DIGITS=0; OPER_STATUS=NF;
TRAFFIC_TYPE=TANDEM; ANI_BASED_ROUTING=N; CLLI=DAL177DS3;
CALL_CTRL_ROUTE_ID=bts12-ccroute1; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;

Reply: Success:

CLI>
```

LNP Examples

The following examples illustrate typical LNP call scenarios.

Example 1

This example illustrates a TVT command on a trunk origination, with CdPN resulting in an LNP query. QVT gets the RN and suggests the second **translate** command. The second TVT shows the route of the outgoing trunk group to the recipient switch.

```
btsadmin> translate trunk tgn-id=5; called-dn=11501160;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->
TABLE: TRUNK_GRP

ID=5; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-3;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=IN; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG IN from Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg5; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=Y;
```

```
TABLE: DIAL_PLAN_PROFILE

.
.
.

TABLE: OFFICE_CODE

DIGIT_STRING=11501; OFFICE_CODE_INDEX=15; DID=N; CALL_AGENT_ID=CA146; DIALAB


LE=Y; NDC=1; EC=150; DN_GROUP=1xxx; EC_DIGIT_STRING=1150;

TABLE: DN2SUBSCRIBER

OFFICE_CODE_INDEX=15; DN=1160; STATUS=PORTED_OUT; RING_TYPE=1; LNP_TRIGGER=N;
NP_RESERVED=N; LAST_CHANGED=2005-08-11 14:30:09.0; VIRTUAL_DN=N; PORTED_IN=N;

 ****** THIS CALL NEEDS AN LNP QUERY ******

****** LNP QUERY is needed (Onward Call Routing query), Suggested QUERY


Command to Run ******

QUERY LNP; tgn-id=5; called-dn=11501160

****** If query result is Routing Number (RN) Not Found,

       the above translation is valid

****** Otherwise, use the TRANSLATE command

       suggested by the query result


Reply: Success:


btsadmin> QUERY LNP tgn-id=5; called-dn=11501160;

QUERY ON FEATURE SERVER FSAIN205 IS... ->

RESULT ->
Called Number=11501160, Routing Number (RN) =4101
**** Suggested TRANSLATE Command ****

TRANSLATE TRUNK tgn_id=5; original_called_dn=11501160; called_dn=4101-11501160;
noa=PORTED_NUMBER_WITH_RN;

btsadmin> TRANSLATE TRUNK tgn_id=5; original_called_dn=11501160; called_dn=4101-11501160;
noa=PORTED_NUMBER_WITH_RN;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->
TABLE: TRUNK_GRP

.
.
.
```

```
ID=inet116_rg1; POLICY_TYPE=ROUTE; POLICY_ID=inet116_rte;

TABLE: ROUTE

ID=inet116_rte; TGN1_ID=6; DEL_DIGITS1=0; DEL_DIGITS2=0; DEL_DIGITS3=0; DEL_DIGITS4=0;
DEL_DIGITS5=0; DEL_DIGITS6=0; DEL_DIGITS7=0; DEL_DIGITS8=0; DEL_DIGITS9=0; DEL_DIGITS10=0;
TG_SELECTION=SEQ; NEXT_ACTION=NONE;

TABLE: TRUNK_GRP

ID=6; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-4;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=OUT; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG OUT to Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg6; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=N;



Reply: Success:
```

Example 2

In this example, a subscriber dials a DN ported-out of this switch. QVT gets the RN, and a second TVT shows the route of the outgoing trunk group to the recipient switch.

Because the called DN is ported-out, the call cannot be routed on this switch without an LNP query. If QVT does not find an RN, perhaps because the DN2RN table is incorrect temporarily during the porting transition, the call will be released due to cause unallocated number.

```
btsadmin> translate line calling-dn=11501511; called-dn=11501160;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sipata1; CATEGORY=INDIVIDUAL; NAME=h15 sipata1 Moe; STATUS=ACTIVE; BILLING_DN=11501511;
DN1=11501511; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501511@192.168.54.124; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;


.
.
.

TABLE: DN2SUBSCRIBER

OFFICE_CODE_INDEX=15; DN=1160; STATUS=PORTED_OUT; RING_TYPE=1; LNP_TRIGGER=N;
NP_RESERVED=N; LAST_CHANGED=2005-08-11 14:30:09.0; VIRTUAL_DN=N; PORTED_IN=N;

 ****** THIS CALL NEEDS AN LNP QUERY ******

****** LNP QUERY is needed (Onward Call Routing query), Suggested QUERY Command to Run
******
```

```
QUERY LNP calling-dn=11501511; called-dn=11501160

****** If query result is Routing Number (RN) Not Found,

       the above translation is valid

****** Otherwise, use the TRANSLATE command

       suggested by the query result


Reply: Success:

btsadmin>
btsadmin>
btsadmin> QUERY LNP calling-dn=11501511; called-dn=11501160

QUERY ON FEATURE SERVER FSAIN205 IS... ->

RESULT ->
Called Number=11501160, Routing Number (RN) =4101
**** Suggested TRANSLATE Command ****

TRANSLATE LINE calling_dn=11501511; original_called_dn=11501160; called_dn=4101-11501160;
NOA=PORTED-NUMBER-WITH-RN;

QUERYSTATUS -> Query Success

Reply: Success:

btsadmin>
btsadmin>
btsadmin> TRANSLATE LINE calling_dn=11501511; original_called_dn=11501160;
called_dn=4101-11501160; NOA=PORTED-NUMBER-WITH-RN;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sipata1; CATEGORY=INDIVIDUAL; NAME=h15 sipata1 Moe; STATUS=ACTIVE; BILLING_DN=11501511;
DN1=11501511; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501511@192.168.54.124; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;


.
.
.


TABLE: TRUNK_GRP
```

```
ID=6; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-4;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=OUT; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG OUT to Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg6; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=N;



Reply: Success:
btsadmin>
```

Example 3

In this example, the first TVT shows a translation but indicates that an LNP query is needed. The QVT does not find an RN, so the first TVT has the correct translation and routing information.

```
btsadmin> translate line calling-dn=11501511; called-dn=11501512;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sipata1; CATEGORY=INDIVIDUAL; NAME=h15 sipata1 Moe; STATUS=ACTIVE; BILLING_DN=11501511;
DN1=11501511; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501511@192.168.54.124; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;

TABLE: SUBSCRIBER_PROFILE

ID=hungary_prof; DIAL_PLAN_ID=dp_sub_itu; LOCAL_PFX1_OPT=NR; TOLL_PFX1_OPT=RQ;
POP_ID=hun1; OLI=0; EA_USE_PIC1=Y; INTERLATA_PFX1_OPT=RQ;


.
.
.

TABLE: SUBSCRIBER

ID=sipata2; CATEGORY=INDIVIDUAL; NAME=h15 sipata2 Larry; STATUS=ACTIVE;
BILLING_DN=11501512; DN1=11501512; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE;
PIC3=NONE; GRP=N; USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP;
IMMEDIATE_RELEASE=N; TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501512@192.168.54.124;
SEND_BDN_AS_CPN=N; SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;

****** LNP QUERY is needed (LNP-TRIGGER for ODBR), Suggested QUERY Command to Run ******
```

```
QUERY LNP calling-dn=11501511; called-dn=11501512

****** If query result is Routing Number (RN) Not Found,

       the above translation is valid

****** Otherwise, use the TRANSLATE command

       suggested by the query result


Reply: Success:

btsadmin>
btsadmin> QUERY LNP calling-dn=11501511; called-dn=11501512

QUERY ON FEATURE SERVER FSAIN205 IS... ->

RESULT ->
Called Number=11501512, Routing Number (RN) Not Found

QUERYSTATUS -> Query Success

Reply: Success:
```

Example 4

This example is for a QOR originating switch. A subscriber dials a DN that is ported-out of another (donor) switch. The call is translated and routed to the donor switch, as shown in the first translate TVT command below. The donor switch sends a REL with LNP QOR: Ported Number cause to the originating switch.

The originating switch receives the REL with LNP QOR: Ported Number cause, and then the originating switch does an LNP query. The QVT query finds an RN, and the RN and NOA are used as input to the TVT to show the routing after the QOR query, as shown in the second translated command below.

```
btsadmin> translate line calling-dn=11501511; called-dn=11161168

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sipata1; CATEGORY=INDIVIDUAL; NAME=h15 sipata1 Moe; STATUS=ACTIVE; BILLING_DN=11501511;
DN1=11501511; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501511@192.168.54.124; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;

TABLE: SUBSCRIBER_PROFILE

ID=hungary_prof; DIAL_PLAN_ID=dp_sub_itu; LOCAL_PFX1_OPT=NR; TOLL_PFX1_OPT=RQ;
POP_ID=hun1; OLI=0; EA_USE_PIC1=Y; INTERLATA_PFX1_OPT=RQ;


.
.
.
```

```
TABLE: TRUNK_GRP

ID=6; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-4;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=OUT; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG OUT to Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg6; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=N;



Reply: Success:

btsadmin>
btsadmin>
btsadmin>
btsadmin>
btsadmin> query LNP calling-dn=11501511; called-dn=11161168;

QUERY ON FEATURE SERVER FSAIN205 IS... ->

RESULT ->
Called Number=11161168, Routing Number (RN) =4001
**** Suggested TRANSLATE Command ****

TRANSLATE LINE calling_dn=11501511; original_called_dn=11161168; called_dn=4001-11161168;
NOA=PORTED-NUMBER-WITH-RN;

QUERYSTATUS -> Query Success

Reply: Success:

btsadmin>
btsadmin>
btsadmin>
btsadmin>
btsadmin> TRANSLATE LINE calling_dn=11501511; original_called_dn=11161168;
called_dn=4001-11161168; NOA=PORTED-NUMBER-WITH-RN;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->

TABLE: SUBSCRIBER

ID=sipata1; CATEGORY=INDIVIDUAL; NAME=h15 sipata1 Moe; STATUS=ACTIVE; BILLING_DN=11501511;
DN1=11501511; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE; PIC3=NONE; GRP=N;
USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP; IMMEDIATE_RELEASE=N;
TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501511@192.168.54.124; SEND_BDN_AS_CPN=N;
SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;

TABLE: SUBSCRIBER_PROFILE

ID=hungary_prof; DIAL_PLAN_ID=dp_sub_itu; LOCAL_PFX1_OPT=NR; TOLL_PFX1_OPT=RQ;
POP_ID=hun1; OLI=0; EA_USE_PIC1=Y; INTERLATA_PFX1_OPT=RQ;


.
.
.
```

```
TABLE: TRUNK_GRP

ID=6; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-4;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=OUT; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG OUT to Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg6; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=N;

Reply: Success:
```

Example 5

This example illustrates an incoming trunk call with an RN prefix and ported number NOA.

Note In this example, the Cisco BTS 10200 reminds you that the NOA and ORIGINAL-CALLED-DN tokens must both be specified.

```
btsadmin> translate trunk tgn-id=5; called-dn=400111501512; NOA=PORTED-NUMBER-WITH-RN;

Reply: Failure: NOA and ORIGINAL-CALLED-DN should be specified together

btsadmin>
btsadmin>
btsadmin> translate trunk tgn-id=5; called-dn=400111501512; NOA=PORTED-NUMBER-WITH-RN;
original-called-dn=11501512;

TRANSLATE ON CALL AGENT CA146 IS... ->

TABLEINFO ->
TABLE: TRUNK_GRP

ID=5; CALL_AGENT_ID=CA146; TG_TYPE=SS7; NUM_OF_TRUNKS=24; DPC=5-2-3;
TG_PROFILE_ID=tgprof_inet116; STATUS=INS; DIRECTION=IN; SEL_POLICY=DSC; GLARE=SLAVE;
ALT_ROUTE_ON_CONG=N; SIGNAL_PORTED_NUMBER=Y; POP_ID=hun1; DIAL_PLAN_ID=dp_trk_itu;
Description=TG IN from Inet 116; DEL_DIGITS=0; TRAFFIC_TYPE=LOCAL; ANI_BASED_ROUTING=N;
CALL_CTRL_ROUTE_ID=cc_rte_i116_tg5; MGCP_PKG_TYPE=T; ANI_SCREENING=N; SEND_RDN_AS_CPN=N;
STATUS_MONITORING=N; SEND_EARLY_BKWD_MSG=N; EARLY_BKWD_MSG_TMR=5; SCRIPT_SUPP=N;
VOICE_LAYER1_USERINFO=AUTO; VOICE_INFO_TRANSFER_CAP=AUTO; ACCESS_TYPE=COMBINED;
POI=INTER_ENDOFFICE; PERFORM_LNP_QUERY=Y;

TABLE: DIAL_PLAN_PROFILE

ID=dp_trk_itu; Description=Trunk Origination Local dial-plan (ITU); NANP_DIAL_PLAN=N;
ANI_DIGMAN_ID=dm_dpp_ani_itu; DNIS_DIGMAN_ID=dm_dpp_trk_itu; OVERDECADIC_DIGITS_SUPP=N;
NOA_BASED_ROUTING=Y; NOA_ROUTE_PROFILE_ID=noa_rt;

TABLE: DIGMAN

ID=dm_dpp_ani_itu; RULE=1; MATCH_NOA=ANY; REPLACE_NOA=NATIONAL;

TABLE: DIGMAN

ID=dm_dpp_trk_itu; RULE=1; MATCH_STRING=^4001; REPLACE_STRING=NONE;
MATCH_NOA=PORTED_NUMBER_WITH_RN; REPLACE_NOA=UNKNOWN;
```

```
TABLE: NOA_ROUTE_PROFILE

ID=noa_rt; Description=NOA Route profile (ITU) to RN dial-plan;

CONTINUE WITH EXISTING DIAL-PLAN

TABLE: DIAL_PLAN

ID=dp_trk_itu; DIGIT_STRING=1150; DEST_ID=dest_sub_itu; SPLIT_NPA=NONE; DEL_DIGITS=0;
MIN_DIGITS=8; MAX_DIGITS=8; NOA=UNKNOWN;

TABLE: DESTINATION

DEST_ID=dest_sub_itu; CALL_TYPE=LOCAL; ROUTE_TYPE=SUB; ZERO_PLUS=N; INTRA_STATE=Y;
Description=ITU Sub dest: Allow LNP query; GAP_ROUTING=N; ANI_DIGMAN_ID=dm_dest_sub_ani;
DNIS_DIGMAN_ID=dm_dest_rn; CLDPTY_CTRL_REL_ALWD=N; CALL_SUBTYPE=NONE;
ACQ_LNP_QUERY=PERFORM_LNP_QUERY;

TABLE: OFFICE_CODE

DIGIT_STRING=11501; OFFICE_CODE_INDEX=15; DID=N; CALL_AGENT_ID=CA146; DIALABLE=Y; NDC=1;
EC=150; DN_GROUP=1xxx; EC_DIGIT_STRING=1150;

TABLE: DN2SUBSCRIBER

OFFICE_CODE_INDEX=15; DN=1512; STATUS=ASSIGNED; RING_TYPE=1; LNP_TRIGGER=Y; NP_RESERVED=N;
SUB_ID=sipata2; LAST_CHANGED=2005-09-08 11:08:47.0; VIRTUAL_DN=N; PORTED_IN=N;

TABLE: SUBSCRIBER

ID=sipata2; CATEGORY=INDIVIDUAL; NAME=h15 sipata2 Larry; STATUS=ACTIVE;
BILLING_DN=11501512; DN1=11501512; PRIVACY=NONE; RING_TYPE_DN1=1; PIC1=NONE; PIC2=NONE;
PIC3=NONE; GRP=N; USAGE_SENS=Y; SUB_PROFILE_ID=hungary_prof; TERM_TYPE=SIP;
IMMEDIATE_RELEASE=N; TERMINATING_IMMEDIATE_REL=N; AOR_ID=11501512@192.168.54.124;
SEND_BDN_AS_CPN=N; SEND_BDN_FOR_EMG=N; PORTED_IN=N; BILLING_TYPE=NONE; VMWI=Y; SDT_MWI=Y;


Reply: Success:

btsadmin>
```

Network Loopback Test for Network-Based Call Signaling/Media Gateway Control Protocol Endpoints

This section describes the feature that provides the capability to perform network loopback tests on any line side PacketCable Network-based Call Signaling protocol specification/Media Gateway Control Protocol (NCS/MGCP) Residential Gateways. The network loopback tests can be initiated from designated test endpoints. This section also describes enhancements to the TDM bearer path test call feature.

This section contains the following:

Overview

The Network Loopback Test for NCS/MGCP Endpoints feature provides a testing device with the capability to perform network loopback tests from any line side NCS/MGCP residential gateways or media termination adapters (MTAs). These loopback tests are initiated from designated test endpoints (subscribers) controlled by the Cisco BTS 10200.

The basic network loopback test feature is service affecting. In other words, while a network loopback call is in progress, the endpoint is considered busy.

The Cisco BTS 10200 network loopback and network continuity tests also have a service-not-affected mode. In this mode, the Cisco BTS 10200 will attempt to create coexisting test connections on the test device; however, if the endpoint does not have enough resources, the Cisco BTS 10200 gives preference to regular calls, processing them first before it processes any test calls.

In the service-affected mode the Cisco BTS 10200 will not try to initiate other calls, even if the MTA/TGW can set up multiple connections (PARALLEL-TEST-CONN-SUPP=Y).

The Cisco BTS 10200 allows the system level configuration to specify whether the network loopback and network continuity test calls will be service affecting or not service affecting.

Restrictions

Although you can test this feature by using the regular MTA as the testing device (by configuring the endpoints as subscriber terminations in Cisco BTS 10200), you need special test equipment such as BRIX if voice quality testing needs to be done.

You should configure the testing and tested devices on the same Call Agent. The Cisco BTS 10200 cannot perform network loopback test calls that originate from another switch and does not route calls from a testing device on an H.323 or SIP interface.

Note You cannot perform the network loopback test if the status of the subscriber to be tested is unequipped (UEQP) or operational-out-of-service (OOS).

Installing

The following items must be configured:

- Test origination endpoints as trunks instead of line
- Special dial plan and destination with CALL-TYPE TEST-CALL; CALL-SUBTYPE=NLB-TEST)

Configuring

In order for parallel test connections to work, the following settings need to be configured in the ca-config:

```
add ca-config type=NLB-TEST-SERVICE-AFFECTING; datatype=BOOLEAN; value=N;
add ca-config type=NCT-TEST-SERVICE-AFFECTING; datatype=BOOLEAN; value=N;
```

Configuration Examples

The following example shows the steps required to configure the originating line (media gateway profile) to identify a network loopback call.

Note These tasks include examples of CLI commands that illustrate how to provision the specific feature. Most of these tables have additional tokens that are not included in the examples. For a complete list of all CLI tables and tokens, see the Cisco BTS 10200 Softswitch CLI Database.

Global Configuration Example

Step 1 Add ca-config NLB-TEST-SERVICE-AFFECTING.

```
add ca-config type=NLB-TEST-SERVICE-AFFECTING; value=N
```

Step 2 Add ca-config NCT-TEST-SERVICE-AFFECTING.

```
add ca-config type=NCT-TEST-SERVICE-AFFECTING; value=N;
```

Step 3 Add ca-config TEST-TRUNK-GRP-DIGITS.

```
add ca-config type=TEST-TRUNK-GRP-DIGITS; value=4;
```

Step 4 Add ca-config TEST-TRUNK-MEMBER-DIGITS.

```
add ca-config type=TEST-TRUNK-MEMBER-DIGITS; value=4;
```

Dedicated NLB Testing Device Configuration Example

The following procedure is a dedicated NLB testing device configuration example. Change TEST-LINE-TYPE to different values (other than NTE) to change test origination type.

Step 1 Add MGW profile.

```
add mgw-profile id=BRIX; vendor=Tollgrade; mgcp-version=mgcp_1_0; MGCP-VARIANT=NCS-1-0;
```

Step 2 Add cas-tg-profile.

```
add cas-tg-profile id=BRIX_TG; sig-type=LINE; TEST-LINE=Y; TEST-LINE-TYPE=NLB-LINE-TEST
```

Step 3 Add MGW.

```
add mgw id=brix1; tsap-addr=<mgw DNS / IP address>; mgw-profile-id=BRIX; type=TGW;
call-agent id=CA146;
```

Step 4 Add trunk-grp.

```
add trunk-grp id=100; call-agent-id=CA146; tg-type=CAS; cas-tg-profile=BRIX_TG;
mgcp-pkg-type=LINE
```

Step 5 Add termination.

```
add termination prefix=aaln/; port-start=1; port-end=2; type=TRUNK; mgw-id=c925.172;
```

Step 6 Add trunk.

```
add trunk termination-prefix=aaln/; termination-port-start=1; termination-port-end=2;
cic-start=1; cic-end=2; tgn-id=100
```

Shared Testing Device Configuration Example

The following procedure is a shared testing device configuration example.

Step 1 Add MGW profile.

```
add mgw-profile id=BRIX; vendor=Tollgrade; mgcp-version=mgcp_1_0; MGCP-VARIANT=NCS-1-0;
```

Step 2 Add cas-tg-profile.

```
add cas-tg-profile id=BRIX_TG; sig-type=LINE; TEST-LINE=Y; TEST-LINE-TYPE=NTE
```

Step 3 Add MGW.

```
add mgw id=brix1; tsap-addr=<mgw DNS / IP address>; mgw-profile-id=BRIX; type=TGW;
call-agent id=CA146;
```

Step 4 Add trunk-grp.

```
add trunk-grp id=100; call-agent-id=CA146; tg-type=CAS; cas-tg-profile=BRIX_TG;
mgcp-pkg-type=LINE
```

Step 5 Add termination.

```
add termination prefix=aaln/; port-start=1; port-end=2; type=TRUNK; mgw-id=c925.172;
```

Step 6 Add trunk.

```
add trunk termination-prefix=aaln/; termination-port-start=1; termination-port-end=2;
cic-start=1; cic-end=2; tgn-id=100
```

Step 7 Add dial-plan-profile.

```
add dial-plan-profile id=dp1; description=NA_Default;
```

Step 8 Add dial-plan.

```
add dial-plan id=dp1; digit-string=919-392; dest-id=sub; noa=national;
```

Step 9 Add digit-map.

```
add digit-map id=test;
digit-pattern=[2-9]xx[2-9]xxxxxx|011xxxxxx.T|01xxxxxx.T|101xxxx|#|*xx|11xx|xxxxxxxxxxxxxxx
xxxx; description=default_pattern
```

Step 10 Add subscriber-profile.

```
add subscriber-profile id=subpf1; digit-map-id=test; dial-plan-id=DP1; POP-ID=1;
```

Step 11 Add subscriber.

```
add subscriber id=sub11; sub-profile-id= subpf1; category=individual; term-id=aaln/0;
mgw-id=c925.172; dn1=919-392-1235; name=RTP5;
```

Tested Line Device Configuration Example

The following procedure is a tested line device configuration example.

Step 1 Add MGW profile.

```
add mgw-profile id=UBR925; vendor=Cisco; mgcp-version=mgcp_1_0; MGCP-VARIANT=NCS_1_0;
```

Step 2 Add MGW.

```
add mgw id=c925.172; tsap-addr=<mgw DNS / IP address>; mgw-profile-id=UBR925; call-agent
id=CA103;
```

Step 3 Add termination.

```
add termination prefix=aaln/; port-start=0; port-end=1; type=line; mgw-id=c925.172;
mgcp-pkg-type=line-ncs;
```

Step 4 Add destination.

```
add destination dest-id=local-call; route-type=sub; call-type=local;
```

Step 5 Add dial-plan-profile.

```
add dial-plan-profile id=dp1; description=NA_Default;
```

Step 6 Add dial-plan.

```
add dial-plan id=dp1; digit-string=919-392; dest-id=sub; noa=national;
```

Step 7 Add subscriber-profile.

```
add subscriber-profile id=subpf1; dial-plan-id=dp1; pop-id=1;
```

Step 8 Add subscriber.

```
add subscriber id=sub11; sub-profile-id= subpf1; category=individual; term-id=aaln/0;
mgw-id=c925.172; dn1=919-392-1235; name=RTP5;
```

Routing for Shared trunk-grp IP Testing Flow Chart Configuration Example

The following procedure is a routing for shared trunk-grp IP testing flow chart configuration example.

Step 1 Add destination.

```
add destination dest-id=DEST_NLB_SUB; call-type=TEST-CALL; call-subtype=NLB-LINE-TEST;
route-type=SUB;

add destination dest-id=DEST_NCT_SUB; call-type=TEST-CALL; call-subtype=NCT-LINE-TEST;
route-type=SUB;

add destination dest-id=DEST_NLB_TRUNK; call-type=TEST-CALL; call-subtype=NLB-TRUNK-TEST;
route-type=ROUTE; route-guide-id=abc

add destination dest-id=DEST_NCT_TRUNK; call-type=TEST-CALL; call-subtype=NCT-TRUNK-TEST;
route-type=ROUTE; route-guide-id=abc
```

Step 2 Add call-subtype-profile.

```
add call-subtype-profile call-type=TEST_CALL; call-subtype=NONE; qos-id=1;
```

Step 3 Add dial-plan-profile.

```
add dial-plan-profile id=test; nanp-dial-plan=N
```

Step 4 Add dial-plan.

```
add dial-plan id=test; digit-string=151; dest-id=DEST_NLB_SUB; min-digits=13;
max-digits=13

add dial-plan id=test; digit-string=152; dest-id=DEST_NCB_SUB; min-digits=13;
max-digits=13

add dial-plan id=test; digit-string=153; dest-id=DEST_NLB_TRUNK; min-digits=13;
max-digits=13

add dial-plan id=test; digit-string=154; dest-id=DEST_NCT_TRUNK; min-digits=13;
max-digits=13
```

Testing Device Status and Control Flowchart Configuration Example

The following procedure is a testing device status and control flowchart configuration example.

Step 1 Control MGW.

```
control mgw id=c925.172; target-state=INS; mode=FORCED;
```

Step 2 Status MGW.

```
status mgw id=c925.172;
```

Step 3 Control trunk-grp.

```
control trunk-grp id=100; call-agent-id=CA146; target-state=INS; mode=forced;
```

Step 4 Equip trunk-termination.

```
equip trunk-termination tgn-id=100; cic=all;
```

Step 5 Control trunk-termination.

```
control trunk-termination tgn-id=100; cic=all; target-state=INS; mode=forced;
```

Step 6 Status trunk-termination.

```
status trunk-termination id=100; cic=all;
```

Step 7 Reset trunk-termination.

```
reset trunk-termination id=100; cic=all;
```

Network Loopback Test for Network-Based Call Signaling/Media Gateway Control Protocol Endpoints

This section describes network loopback testing for network-based call signaling and media gateway control protocol endpoints feature and includes descriptions of the following:

- Dedicated Test Trunk Group
- Shared Test Trunk Group
- Configuring the Originating Trunk Group

To use this feature, place a call from the testing device subscriber to any MGCP subscriber to be tested. For example, if the testing device is an MGCP telephone, dial the number of the subscriber to be tested.

Dedicated Test Trunk Group

The Cisco BTS 10200 allows NCS/MGCP endpoints in a trunk group to be provisioned as a dedicated test trunk group.

The provisioning of the test trunk group determines if incoming calls arriving on the dedicated test trunk groups trigger the Cisco BTS 10200 to complete the test call through a Network Loopback (NLB) or Network Continuity Test (NCT). The category of the test call is preprovisioned on the dedicated test trunk groups—all calls from a particular test trunk group invoke the same test category while calls from another test trunk group might invoke a different test category. A test call from a test device utilizes the eMTA directory number (DN) the same as any other regular dialed digit string.

The called party number format is:

<Test-data>

Where:

<Test-data> = DN (for example, the NCS/MGCP dialed digits signaled to the Cisco BTS 10200 are in the form of a 10-digit DN such as 2145261234, or <TG>TM> (Trunk group and trunk member)

The steps for configuring the originating trunk group are

Step 1 Add a trunk group for the testing device as CAS trunk group (TRUNK-GRP::TG-TYPE=CAS).

Step 2 Associate the trunk group to CAS-TG-PROFILE specific to network loopback test origination type (CAS-TG-PROFILE::TEST-LINE=Y; CAS-TG-PROFILE::TEST-LINE-TYPE=NLB-LINE/NCT-LINE/NLB-TRUNK/NCT-TRUNK.

Step 3 Add all test lines in the testing device as trunk termination.

Shared Test Trunk Group

In addition to dedicated test trunk groups, the Cisco BTS 10200 allows a shared test trunk group, where the category of the test to be run is specified by the test-prefix. Cisco BTS 10200 allows a test trunk group to be associated with a test dial plan. The test trunk group can be either the IP or CAS TDM trunks. Incoming calls from the network on these trunk groups are analyzed according to a preconfigured test dial plan. The following is the format of dialed digits for these incoming test calls.

Called party number format:

<Test-prefix><Test-data>

Where:

- **<Test-prefix>** is a string of digits that denote the test category. Operator must configure the definition (recommended as a pattern of 1 to 6 digits, the Cisco BTS 10200 Softswitch will perform the longest match) of the test prefix and its length, whether it is IP or TDM testing. If it is TDM testing, the traditional 1xx test type value is expected or the general TDM test category needs to be specified (for example, 199) when the route out DN testing is going to be used.

  For example, test-prefix 152 may denote NLB IP testing, or 105 may convey the TDM 105 test-type, or 199 may be defined to specify the TDM route out DN testing, or 153 is the configured prefix for NCT.
- **<Test-data>** is a string that depends on the test-prefix content.

Configuring the Originating Trunk Group

The following are the steps for configuring the originating trunk group:

Step 1 Add a trunk-group for the testing device as CAS trunk-group (TRUNK-GRP::TG-TYPE=CAS).

Step 2 Associate the trunk-grp to CAS-TG-PROFILE specific to network loopback test origination (CAS-TG-PROFILE::TEST-LINE=Y; CAS-TG-PROFILE::TEST-LINE-TYPE=NTE.

Step 3 Configure all test lines in Testing device as trunk-termination.

Step 4 Configure the test dial plan destination with the exact type of test call.

Step 5 Configure the call subtype profile.

Step 6 Configure the main subscriber ID for testing trunk-grp.

Step 7 Configure the digit map for collecting prefixed digits and associate it to the SUBSCRIBER-PROFILE table.

Session Initiation Protocol Subscriber Registration Status Check

The SIP subscriber registration status check CLI command (sip-reg-contact) is used to check the registration status of a SIP subscriber. The need to check the registration status of a SIP subscriber can arise, for example, when a subscriber complains about not being able to receive calls. The first item to check would be the registration status; use the **sip-reg-contact** command. The next item would be to check for events regarding authentication failures and so on.

The following examples show the usage of the **sip-reg-contact** command. The first example shows an expired contact and the second example shows a registered contact or current contact.

Example 1:

Use CLI to check the registration status of an address of record (AOR).

```
CLI> status sip-reg-contact
CLI> AOR_ID=4692551119@sia-SYS44CA146.ipclab.cisco.com;
AOR ID -> 4692551119@sia-SYS44CA146.ipclab.cisco.com
USER -> 4692551119
HOST -> 10.89.220.21
PORT -> 5060
USER TYPE -> USER_PHONE_TYPE
EXPIRES -> 3600
EXPIRETIME -> Tue Oct 7 12:13:11 2003
STATUS -> EXPIRED CONTACT
Reply: Success:
```

Example 2:

Use CLI to check the registration status of an AOR.

```
CLI> status sip-reg-contact
CLI> AOR_ID=4692551001@sia-SYS44CA146.ipclab.cisco.com;
AOR ID -> 4692551001@sia-SYS44CA146.ipclab.cisco.com
USER -> 4692551001
HOST -> 10.89.223.193
PORT -> 5060
USER TYPE -> USER_IP_TYPE
EXPIRES -> 3600
EXPIRETIME -> Thu Oct 23 16:23:48 2003
STATUS -> REGISTERED CONTACT
Reply: Success:
```

System Health Report

The System Health Report (system-health) (SHR) allows the retrieval of the status of various processes within the Cisco BTS 10200.

Use the following example shows you how to run a SHR:

```
CLI> report system-health period=720;
```

Period	The length of time to collect in hours. INTEGER: 1–720 (Default = 24).

The **SHR** command can be used in conjunction with the command scheduler. Using the command scheduler, the SHR runs at periodic intervals collecting the last 24 hours (configurable) worth of data. Upon initial installation and startup of the Cisco BTS 10200, an **SHR** command is scheduled to execute at midnight every 24 hours.

To schedule multiple **SHR** command(s) at different times, you can use the **command scheduler add** command multiple times:

```
CLI> add scheduled-command verb=report; noun=system-health; <recurrence=daily>;
<start-time=...>; <keys=period>; <values=...>
```

Use the following command to remove any scheduled **SHR** command(s):

```
CLI> delete scheduled-command id=NNN
```

Use the following command to obtain an ID number and view the list of scheduled command(s):

```
CLI> show scheduled-command verb=report; noun=system-health
```

To reschedule an **SHR** command for another time, change the recurrence, or change the collection period, use the following command:

```
CLI> change scheduled-command id=NNN; <recurrence=daily>; <start-time=...>; <keys=period>;
<values=...>
```

Fast Audit and Sync Tool

The bts_audit and bts_sync process tools involve running two commands, bts_audit and bts_sync. The bts_audit and bts_sync tools are designed to improve speed and integrity of auditing and syncing the Cisco BTS 10200 databases. The tools can audit and synchronize all mismatches between network elements.

These tools are not a part of the CLI, but are UNIX programs that are run by the root user. They bypass the platform messaging paths and access the EMS, CA, FSPTC, and FSAIN databases directly using database tools. The data is manipulated and updates are applied directly to synchronize the databases.

The bts_audit tool is able to

- Find tables with mismatches
- Find rows missing in application database
- Find rows missing in EMS database
- Find rows with data mismatches between two databases
- Generate a report that lists these mismatches
- Generate the SQL to be used to correct the mismatches

The bts_sync tool is used to send the generated SQL statements to the appropriate destination to bring the databases into synchronization.

The Cisco BTS 10200 fast audit and sync tools feature consists of two UNIX shell scripts that use other UNIX scripts and utilities to perform full-database and table audits of the databases on the various network elements of the system. The database mismatches are synchronized using the bts_sync tool.

The bts_audit tool determines the table sizes when performing full database audit by analyzing the catalog of the CA, FSPTC and FSAIN databases. The scripts will create copies of the data from the tables in a standardized format. The data files are used to generate a checksum for each table. The checksums

are compared, and if they are not equal, the network element data file is transferred to the EMS. On the EMS, the data is compared row by row, and mismatches are printed to a file that can be used by the bts_sync tool to restore synchronization of the table on the network element.

Restrictions and Limitations

The Cisco BTS 10200 fast audit and sync tools feature has the following restrictions and limitations:

- The bts_audit/bts_sync tools are unable to audit and synchronize certain scenarios, such as when a termination record points to an invalid mgw.
- The bts_sync tool should only be run to synchronize the data mismatches between the active platforms.
- If an audit is given a list of tables, and a table references a missing row in another table, the mismatch will not be resolved by the sync.

Using the bts_audit Tool

To use the bts_audit tool, log in at the UNIX root prompt and execute the **bts_audit** command.

Using the bts_sync Tool

To use the bts_sync tool, the **bts_audit** command must be executed first. Log in at the unix root prompt and execute the **bts_audit** command. Once the **bts_audit** command is execution is complete, execute the **bts_sync** command to synchronize the system databases.

Command Parameters

This section describes the parameters for the **bts_audit** and **bts_sync** commands. The following is an example of the **bts_audit** command parameters:

Example:

```
bts_audit -ems <ems> -ca <ca> [-platforms <platforms>] [-tables <tables>]
```

Where:

ems is the hostname of the active EMS machine.

ca is the hostname of the active CA machine.

platforms is a list of the platforms to be audited without spaces and separated by commas

tables is a list of tables to be audited without spaces and separated by commas.

Example:

```
bts_audit -ems priems01 -ca prica01 -platforms CA146,FSAIN205 -tables
SUBSCRIBER,MGW_PROFILE
```

The **bts_sync** command takes a list of filenames to be used for correcting errors found by the audit.

Example:

```
bts_sync /opt/ems/report/Audit_CA146_root.sql
```

or

```
bts_sync /opt/ems/report/Audit_*_root.sql
```

Command Responses

The execution of the **bts_audit** command will output a list of database mismatches found.

Database Out of Synchronization

To troubleshoot database out of synchronization alarms, take the following steps:

Step 1 Log in the system at the unix root prompt.

Step 2 Execute the **bts_audit** command.

Step 3 Once the audit is completed, execute the **bts_sync** command.

IDX Database Auditing

BTS 10200 provides the functionality to compare (audit) the IDX database on both the CA/FS nodes. The IDX database is a proprietary shared memory database. Previous versions of BTS 10200 provided audit functionality between EMS to EMS and EMS to CA/FS nodes. The IDX database auditing feature enables comparison of IDX DB between CA/FS to CA/FS nodes.

There are two kinds of tables in IDX database.

- Static table—contains provisioned data from EMS. Provisioned data is generally static data. Some static tables may also have call-related dynamic data.
- Dynamic table—call related data. This data could be changing quickly and dynamically while calls are being set up, established and terminated. The dynamic tables also contain replication queue information, which holds the replication requests.

Note that this feature audits only the static tables because dynamic tables contain fast changing data, which is difficult to audit.

This feature uses the PAS (Platform Application Server) interface. PAS is the platform independent inter-node XML messaging interface. The messaging is through SSL over TCP/IP connection. Each platform on CA/FS node, CA, FSAIN and FSPTC, has a PAS server process taking requests and sending back responses.

dbm_audit

The dbm_audit program is the executable file that user needs to execute on any EMS node to perform CA/FS to CA/FS IDX audit. This feature is executed from the Unix command shell on any EMS node. The platform status of CA/FS should be either Active or Standby to perform this IDX database audit.

The executable file of this feature is named *dbm_audit*, and it is installed in the */opt/bts/bin* directory on all BTS nodes. The audit report is generated in the */opt/ems/report* directory on the EMS node.

The tool compares the following:

- The table row count
- The index of each record
- The content of each record

The file name of the audit report contain the following:

- The hostname of both CA/FS nodes
- The time stamp of the audit complete time

The audit report contains the following information:

- Audit starting time
- Audit ending time
- The error message if audit fails
- Number of the mismatched tables
- Entry for every audited table
  - Name of the audited table
  - The application (platform) that associates with the audited table
  - Overall audit result of the audited table
  - Type of the mismatch if any, in the row count, index, record content
  - The index number of the mismatched IDX record

Command Parameters

The following is the command usage for using *dbm_audit* executable:

dbm_audit -table <*value1***> -platform <***value2***> -type <***value3***>**

where the -**table** parameter can have the following values:

- **all**
- a static table name

The "**all**" value is the default value if the -**table** parameter is not specified.

where the -**platform** parameter can have the following values:

- **CA**
- **FSAIN**
- **FSPTC**
- "**all**" for all the above platforms.

The "**all**" value is the default value if the -**platform** parameter is not specified.

where the -**type** parameter can have the following values:

- **full**
- **row_count**

The "**row_count**" value is the default value if the -**type** parameter is not specified.

Error Conditions

The *dbm_audit* tool displays the following error message when the audit is performed between IDX databases of different release versions of BTS 10200:

```
ERROR: Unable to audit different release DB
```

The *dbm_audit* tool skips the audit of a platform if the status of the platform is neither active or standby. Following error message is displayed:

```
ERROR: <CA/FS_appname> on <hostname> is neither Active nor Standby
```

The *dbm_audit* tool exits with the following error message when a dynamic table name is given for table audit option:

```
WARNING: <table_name> is not a static table in <platform_name>
```

The *dbm_audit* tool exits with the following error message when at least one of the CA/FS transitions to OOS status during audit.

```
ERROR: <CA/FS_appname> going OOS
```

The *dbm_audit* tool exits gracefully with the following error message when the tool is unable to set up PAS connection to the CA/FS nodes.

```
ERROR: Platform status is found OOS for CA-B on <hostname>
ERROR: Unable to perform DBM audit on CA
```

The *dbm_audit* tool exits gracefully with the following error message when the PAS session times out:

```
ERROR: PAS session not responding
```

ISDN Network Loopback Test

This section describes the Network Loopback (NLB) Test for ISDN PRI trunks (ISDN NLB) feature. Network Loopback Test for ISDN-PRI trunks (ISDN NLB) feature allows operators to conduct network loopback testing originating from shared ISDN PRI trunks. The shared test trunk group accepts both normal and test calls. Test calls are identified by provisioning the call-type and call-subtype tokens in the Destination table.

The Cisco BTS 10200 cannot perform network loopback test calls that originate from another switch and does not route calls from a testing device on an H.323 or SIP interface.

Note The network loopback test cannot be performed if the status of the subscriber to be tested is unequipped (UEQP) or operational-out-of-service (OOS).

Configuring

The following items must be configured:

- Test origination endpoints as trunks instead of lines.
- Special dial plan and destination with call-type=test-call.

- Call-subtype must be configured as one of:
  - nlb-line-test
  - nct-line-test
  - nlb-trunk-test
  - nct-trunk-test

Originating Trunk Group

The ISDN NLB feature uses a shared test trunk group, where the type of test is specified by the test-prefix. Cisco BTS 10200 allows a test trunk group to be associated with a test dial plan. The test trunk group is an ISDN PRI trunk. Incoming calls from the network on an ISDN PRI trunk are analyzed according to a preconfigured test dial plan. The following is the format of dialed digits for these incoming test calls.

Called party number format:

<Test-prefix> <Test-data>

Where:

- <Test-prefix> is a string of digits that denote the test category. Operator must configure the definition of the test prefix and its length. We recommend a pattern of 1 to 6 digits—but the first digit cannot be "1", the Cisco BTS 10200 performs the longest match.
- <Test-data> is a string that depends on the test-prefix content. The following steps configure the originating trunk group:

Step 1 Add a trunk-group for the testing device as an ISDN PRI trunk-group if it does not already exist.

```
trunk-grp tg-type=isdn;
```

Step 2 Configure the test dial plan destination with the exact type of test call.

Step 3 Configure the call type subprofile.

Step 4 Configure a main subscriber ID for the testing trunk group if necessary.

Call Agent Configuration Table

The system defaults for the Call Agent Configuration (ca-config) table may require changing, based on the needs of the test. Take the following steps to change the service affect of the test.

Step 1 Execute the following commands to change service affect for either NCT or NLB testing. The default service affect is Y.

```
change ca-config nct-test-service-affecting=n;
change ca-config nlb-test-service-affecting=n;
```

  - Y—Subscriber under test cannot make or receive calls.
  - N—Subscriber under test can make or receive calls; test calls are dropped.

Step 2 Define the number of digits for the trunk group and CICs that are under test. The defaults for both are 4.

```
change ca-config test-trunk-grp-digits=<x>;
change ca-config test-trunk-member-digits=<x>;
```

Dial Plan

If the nanp-dial-plan token in the Dial Plan Profile table is set to Y, then the nature of address (NOA) in the Dial Plan table cannot be unknown. The NOA can be set to national. The first digit of the prefix cannot be 1—use any number between 2 and 9.

Sample Configurations

The following sample configurations illustrate how to configure the Cisco BTS 10200 for ISDN NLB with network terminating equipment (NTE).

Note In these samples, digit-string=*nnn* (where nnn = 551 and so forth), *nnn* is the test-prefix.

Note These tasks include examples of CLI commands that illustrate how to provision the specific feature. Most of these tables have additional tokens that are not included in the examples. For a complete list of all CLI tables and tokens, see the Cisco BTS 10200 Softswitch CLI Database.

Line Loopback Tests Over an ISDN Trunks

This section provides examples of Network Test Equipment (NTE) line loopback over ISDN trunks.

NLB Tests

This section provides examples of network loopback (NLB) line loopback tests over ISDN trunks.

NLB Line Loopback Test Over an ISDN Trunk

This section provides sample steps for the NTE NLB trunk test over an ISDN trunk feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nlb-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial** plan id=<xxx>; digit-string=551; dest-id=nlb-trunk-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 551) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.

	Perform the Following Command or Action:	Purpose and Comments
Step 4	From the test equipment, dial the NTE NLB trunk test call (551+xxx-xxx-xxxx)	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 5	Hang up the test call and verify the Billing call type.	—

NLB Line Loopback Test Over an ISDN Trunk With Service Affecting Turned On

This section provides sample steps for the NTE NLB line test over an ISDN trunk with "service affecting" turned on feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nlb-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=551; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 551) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=y; datatype=boolean; value=y;	Provision the Call Agent Configuration table with service affecting on.
Step 5	From the test equipment, dial the NTE NLB line test call (551+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 6	Take the subscriber under test off-hook.	There is no dial tone.
Step 7	Call the subscriber under test from another subscriber.	Call is treated, and the test call is still active.
Step 8	Hang up the test call and verify the Billing call type.	—

NLB Line Loopback Test Over an ISDN Trunk With Service Affecting Turned Off and Parallel Test Connection Support Turned Off

This section provides sample steps for the NTE NLB line test over an ISDN trunk with "service affecting" turned off feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nlb-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.

	Perform the Following Command or Action:	Purpose and Comments
Step 3	**add dial plan** id=<xxx>; digit-string=551; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 551) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	From the test equipment, dial the NTE NLB-LINE test call (551+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 6	Take the subscriber under test off-hook.	There is a dial tone.
Step 7	Call the subscriber under test from another subscriber.	Call is set up, and the test call is released.
Step 8	Hang up the test call and verify the Billing call type.	—

NLB Line Loopback Test Over an ISDN Trunk With Service Affecting Turned Off and Parallel Test Connection Support Turned On: Call from Subscriber Under Test

This section provides sample steps for the NTE NLB line test over an ISDN trunk with "service affecting" turned on and parallel test connection support turned on feature. The call is from the subscriber under test.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nlb-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=551; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 551) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	**change mgw-profile** id=isdnNLB; parallel-test-conn-supp=y;	Turn on support parallel test connection in the Media Gateway Profile table.
Step 6	From the test equipment, dial the NTE NLB line test call (551+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 7	Take the subscriber under test off-hook.	There is a dial tone.
Step 8	Call the subscriber under test from another subscriber.	Call is set up, and the test call is still active.
Step 9	Hang up the test call and verify the Billing call type.	—

NLB Line Loopback Test Over an ISDN Trunk With Service Affecting Turned Off and Parallel Test Connection Support Turned On: Call to Subscriber Under Test

This section provides sample steps for the NTE NLB line test over an ISDN trunk with "service affecting" turned off and parallel test connection support turned on feature. The call is to the subscriber under test.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nlb-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=551; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 551) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	**change mgw-profile** id=isdnNLB; parallel-test-conn-supp=y;	Turn on support parallel test connection in the Media Gateway Profile table.
Step 6	From the test equipment, dial the NTE NLB-LINE test call (551+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 7	Take the subscriber under test off-hook.	There is a dial tone.
Step 8	Call the subscriber under test from another subscriber.	Call is set up, and the test call stays up.
Step 9	Verify the Billing call type.	—

NCT Tests

This section provides examples of line loopback network continuity tests (NCT) over ISDN.

NCT Line Loopback Test Over an ISDN Trunk

This section provides sample steps for the NTE NLB trunk test over an ISDN trunk feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nct-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.

	Perform the Following Command or Action:	Purpose and Comments
Step 3	**add dial plan** id=<xxx>; digit-string=552; dest-id=nlb-trunk-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 552) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	From the test equipment, dial the NTE NLB trunk test call (552+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 5	Hang up the test call and verify the Billing call type.	—

NCT Line Loopback Test Over an ISDN Trunk With Service Affecting Turned On

This section provides sample steps for NTE NCT line test over an ISDN trunk with "service affecting" turned on feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nct-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=552; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 552) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=y; datatype=boolean; value=y;	Provision the Call Agent Configuration table with service affecting on.
Step 5	From the test equipment, dial the NTE NLB line test call (552+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 6	Take the subscriber under test off-hook.	There is no dial tone.
Step 7	Call the subscriber under test from another subscriber.	Call is treated, and the test call is still active.
Step 8	Hang up the test call and verify the Billing call type.	—

NCT Line Loopback Test Over an ISDN Trunk With Service Affecting Turned Off and Parallel Test Connection Support Turned Off

This section provides sample steps for the NTE NCT line test over an ISDN trunk with "service affecting" turned off feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nct-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=552; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match to the LB test prefix (for example 552) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	From the test equipment, dial the NTE NLB-LINE test call (552+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 6	Take the subscriber under test off-hook.	There is a dial tone.

	Perform the Following Command or Action:	Purpose and Comments
Step 7	Call the subscriber under test from another subscriber.	Call is set up, and the test call is released.
Step 8	Hang up the test call and verify the Billing call type.	—

NCT Line Loopback Test Over an ISDN Trunk with Service Affecting Turned Off and Parallel Test Connection Support Turned On: Call from Subscriber Under Test

This section provides sample steps for the NTE NCT line test over an ISDN trunk with service "affecting turned" on and parallel test connection support turned on feature. The call is from the subscriber under test.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nct-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=552; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 552) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	**change mgw-profile** id=isdnNLB; parallel-test-conn-supp=y;	Turn on support parallel test connection in the Media Gateway Profile table.
Step 6	From the test equipment, dial the NTE NLB line test call (552+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 7	Take the subscriber under test off-hook.	There is a dial tone.
Step 8	Call the subscriber under test from another subscriber.	Call is set up, and the test call is still active.
Step 9	Hang up the test call and verify the Billing call type.	—

NCT Line Loopback Test Over an ISDN Trunk With Service Affecting Turned Off and Parallel Test Connection Support Turned On: Call to Subscriber Under Test

This section provides sample steps for the NTE NCT line test over an ISDN trunk with "service affecting" turned off and parallel test connection support turned on feature. The call is to the subscriber under test.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-line-test; call-type=test-call; call-subtype=nct-line-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.

	Perform the Following Command or Action:	Purpose and Comments
Step 3	**add dial plan** id=<xxx>; digit-string=552; dest-id=nlb-line-test; split-npa=none; del-digits=0; min-digits=13; max-digits=13; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match to the LB test prefix (for example 552) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add ca-config** type=nlb-test-service-affecting=n; datatype=boolean; value=n;	Provision the Call Agent Configuration table with service affecting off.
Step 5	**change mgw-profile** id=isdnNLB; parallel-test-conn-supp=y;	Turn on support parallel test connection in the Media Gateway Profile table.
Step 6	From the test equipment, dial the NTE NLB-LINE test call (552+xxx-xxx-xxxx).	The BCM does not notify the Feature Server of this call and the call is looped back.
Step 7	Take the subscriber under test off-hook.	There is a dial tone.
Step 8	Call the subscriber under test from another subscriber.	Call is set up, and the test call stays up.
Step 9	Verify the Billing call type.	—

Trunk Loopback Tests Over an ISDN Trunk

For trunk loopback testing when the test call and normal call are on the same circuit, the normal call always has precedence. For example:

1. If the test call is on circuit *xxx* and a normal call comes in on the same circuit, then the normal call is set up and the test call is released.
2. If a normal call is on circuit *xxx* and a test call comes in on same circuit, then the normal call stays up and the test call is released.

NLB Trunk Loopback Test Over an ISDN Trunk

This section provides sample steps for the NTE NLB trunk test over an ISDN trunk feature.

	Perform the Following Command or Action:	Purpose and Comments
Step 1	**add destination** dest-id=nlb-trunk-test; call-type=test-call; call-subtype=nlb-trunk-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=553; dest-id=nlb-trunk-test; split-npa=none; del-digits=0; min-digits=11; max-digits=11; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 553) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.

	Perform the Following Command or Action:	Purpose and Comments
Step 4	From the test equipment, dial the NTE NLB trunk test call (553+trunk digits+members).	—
Step 5	Hang up the test call and verify the Billing call type.	—

NCT Trunk Loopback Test Over an ISDN Trunk

This section provides sample steps for the NTE NLB trunk test over an ISDN trunk feature.

	Perform the Following Command or Action:	Purpose
Step 1	**add destination** dest-id=nct-trunk-test; call-type=test-call; call-subtype=nct-trunk-test;	Provision the Destination table.
Step 2	**add call-subtype-profile** call-type=TEST_CALL; call-subtype=NONE; qos-id=1;	Provision the Call Subtype Profile table.
Step 3	**add dial plan** id=<xxx>; digit-string=554; dest-id=nlb-trunk-test; split-npa=none; del-digits=0; min-digits=11; max-digits=11; noa=national; **Note** Where <xxx> is an existing dial plan. The dial plan id must match the LB test prefix (for example 554) in the digit string.	Provision the Dial Plan table. The digit-string plus the min-digits and max-digits total depends on the settings configured (if any) in the "Call Agent Configuration Table" section.
Step 4	**add trunk-grp** id=nte; call-agent-id=CA146; tg-type=isdn; dial-plan-id=nte; dpc=101-55-103; tg-profile-id=ISDN1; call-ctrl-route-id=ccr1;	Provision the Trunk Group table.
Step 5	From the test equipment, dial the NTE NLB trunk test call (554+trunk digits+members) (trunk).	—
Step 6	Hang up the test call and verify the Billing call type.	—

Enhanced Traffic Measurement

The Cisco BTS 10200 supports traditional PSTN measurements as well as additional requirements demanded by the IP and ATM backbones over which the services are offered. Many of the informational elements within the measurement data find their basis in the traditional PSTN TDM network implementations with modifications and additions caused by the expanded needs and capabilities of the converged network environment. The Cisco BTS 10200 measurement information includes both statistical and performance details. The mechanism used to manage the data generated and transported from the Cisco BTS 10200 system follows legacy type procedures and is documented in the following sections.

Measurement Data Transport and Access

The measurement data collected on the Cisco BTS 10200 can be accessed through several different mechanisms. The Command Line Interface, which runs over a telnet or SSH session, is used in the examples within this document. Measurement data is also available in CSV or XML format through the

FTP or SFTP interface. The measurement data can be provisioned and is accessible through the SNMP MIB. The supported version of SNMP on the Cisco BTS 10200 is v2c. There is detailed information on both of these access mechanisms available in separate operations manuals.

Measurement Data Event Reports

The measurement subsystem within the Cisco BTS 10200 supports several events that are issued in various abnormal scenarios. Table 15-5 illustrates the event reports that the measurements subsystem supports and their significance.

Table 15-5 Event Reports Supported by Measurement Subsystem

Type and Number	Severity	Description	Meaning
Statistics (2)	Informational	Call Agent Measurement Collection Started	Issued whenever the traffic process running on the call agent platform begins a new collection cycle for the current interval
Statistics (3)	Informational	Call Agent Measurement Collection Finished	Issued whenever the traffic process running on the call agent platform completes a collection cycle for the current interval
Statistics (4)	Informational	POTS/CTX/TDM Measurement Collection Started	Issued whenever the traffic process running on the POTS Feature Server platform begins a new collection cycle for the current interval
Statistics (5)	Informational	POTS/CTX/TDM Measurement Collection Finished	Issued whenever the traffic process running on the POTS Feature Server platform completes a collection cycle for the current interval
Statistics (6)	Informational	AIN Measurement Collection Started	Issued whenever the traffic process running on the AIN Feature Server platform begins a new collection cycle for the current interval
Statistics (7)	Informational	AIN Measurement Collection Finished	Issued whenever the traffic process running on the AIN Feature Server platform completes a collection cycle for the current interval
Statistics (8)	Warning	Message Send Failure	Issued whenever the traffic manager process in the EMS or the traffic agent process in any element is unable to send an inter-process message
Statistics (9)	Warning	Measurement Table SQL Read Error	Issued whenever the traffic manager process in the EMS is unable to read from one of the measurement tables stored in Oracle
Statistics (10)	Warning	Measurement Table SQL Write Error	Issued whenever the traffic manager process in the EMS is unable to write to one of the measurement tables stored in Oracle

Table 15-5 Event Reports Supported by Measurement Subsystem (continued)

Type and Number	Severity	Description	Meaning
Statistics (11)	Warning	Measurement Collection API Failure	Issued whenever the traffic agent process in any of the Cisco BTS 10200 elements is unable to access the counter stored within shared memory by means of the standard API invocations
Statistics (12)	Major	Schemas out of Synchronization	Issued whenever system detects a mismatch between the counter schema in Oracle on the BDMS and the internal schema of the call agents and/or feature servers
Statistics (13)	Major	TMM API Failure	Issued whenever the TMM collection process is unable to initialize or attach to shared memory
Statistics (14)	Warning	MDII Trunk	Calls on the MDII trunk termination are not being successfully completed
Statistics (15)	Minor	Threshold Crossing Alert	A threshold crossing has occurred

Operating

The following sections provide detailed information on how to manage and control the measurement information generated by the Cisco BTS 10200 system. Actual examples are provided with explanations to illustrate the operational mechanics. These and other commands are documented in the Cisco BTS 10200 Softswitch CLI Database and the *Cisco BTS 10200 Operations and Maintenance Guide*.

Provisioning Measurement Report Types

The Cisco BTS 10200 system provides a command line interface to manage the collection of the measurement information generated. This mechanism provides the ability to enable or disable the collection of measurement data and specify the reporting interval on a per report type basis. The factory default setting is to enable the collection of all measurement types and to set the reporting intervals to 15 minutes. Currently, there are 25 types of measurement data generated by the Cisco BTS 10200 (see the following list):

- ISDN—ISDN signaling protocol related information
- CALLP—Call Processing specific information
- MGCP—MGCP signaling protocol related information
- SIM—Service Interaction Manager related information
- POTS-SVC—POTS/Centrex/Tandem Feature Service related information
  - POTS-LOCAL—Local Feature counters
  - POTS-MISC—Miscellaneous Feature counters
  - POTS-SLE—Screening List Editing counters
  - POTS-ACAR—Auto Callback / Recall counters
  - POTS-COS—Class Of Service counters
  - POTS-COT—Customer Originated Trace counters

- AINSVC—AIN Feature Service related information
- ISUP—ISDN User Part (SS7) signaling protocol related information—in a Signaling Gateway configuration
- AUDIT—Auditing related information
- SIA—SIP Interface Adapter related information
- BILLING—Call Detail Data related information
- EM—Event Messaging Billing related information
- DQOS—Dynamic Quality of Service related information
- SNMP—SNMP agent protocol related information
- TG-USG—Trunk Group usage information
- ANM—Announcement server related information
- H323—H.323 signaling protocol related information
- M3UA—M3UA signaling protocol related information
- SUA—SUA signaling protocol related information
- SCTP—SCTP signaling protocol related information
- SCCP—SCCP protocol related information
- TCAP—TCAP related protocol information
- CALL-TOOLS—Metrics related to invocations of the Translation Verification Tools
- AIN-TOOLS—Metrics related to invocations of the Toll Free and LNP Query Verification Tools
- PCT-TOOLS—Metrics related to invocations of the LIDB Query Verification Tools
- ALL—All categories of measurements available on the Cisco BTS 10200

The following is an example of the command line used to provision the collection of the call processing measurement data:

```
change measurement-prov type=callp; enable=yes; time-interval=15;
```

The following is a list of the command line tokens associated with this command and the valid values and purpose of each:

- Type—An ASCII character string from 3 to 8 characters long. The string must match one of the types listed above. This is a mandatory token.
- Enable—An ASCII character string of Yes or No. This string specifies whether or not to perform collection on the specified measurement type. This is an optional token that is preprovisioned with a value of yes at the factory. Either this token and/or the time-interval token must be entered.
- Time-interval—A decimal value of 5, 15, 30, or 60. This value indicates the number of minutes each reporting interval is to encompass for the given report type. The reporting interval is always synchronized to 0 minutes after the hour for consistency. This is an optional token that is preprovisioned with a value of 15 at the factory. Changing this value does not take effect until the completion of the current collection interval based on the previous time-interval setting. Either this token and/or the enable token must be entered.

The following are examples of the command line invocations to display the current settings for the data described above:

```
show measurement-prov type=callp;

show measurement-prov type=anm;
```

```
show measurement-prov type=isdn;

show measurement-prov type=billing;
```

```
show measurement-prov type=em;

show measurement-prov type=snmp;

show measurement-prov type=mgcp;

show measurement-prov type=sim;

show measurement-prov type=pots-fs;

show measurement-prov type=ainsvc;

show measurement-prov type=tcap;

show measurement-prov type=m3ua;

show measurement-prov type=sua;

show measurement-prov type=sctp;

show measurement-prov type=sccp;

show measurement-prov type=isup;

show measurement-prov type=audit;

show measurement-prov type=sia;

show measurement-prov type=dqos;

show measurement-prov type=tg-usg;

show measurement-prov type=h323;

show measurement-prov type=call-tools;

show measurement-prov type=ain-tools;

show measurement-prov type=pct-tools;
```

Measurement Report Summaries

The Cisco BTS 10200 system provides a command line interface (CLI) command for querying summary reports of measurement data from the database on the Element Management System (EMS). This mechanism provides the ability for specifying an interval and the particular type and source of data. The time interval specified must be prior to the current collection interval.

The following are examples of the command line queries to generate reports on the various types of measurements collected from the designated call agents and feature servers from 10 am until noon on March 27th, 2007 and place the data into CSV files for FTP.

Note Any measurement counters that do not contain data for a given interval are kept out of the generated reports. Only counters that were pegged are presented in the resulting summaries.

```
report measurement-isdn-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=isdn-report; output-type=csv;
```

```
report measurement-callp-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id= CA146; output=callp-report; output-type=csv;

report measurement-mgcp-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id= CA146; output=mgcp-report; output-type=csv;

report measurement-sim-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id= CA146; output=sim-report; output-type=csv;

report measurement-pots-local-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-local-report; output-type=csv;

report measurement-pots-misc-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-misc-report; output-type=csv;

report measurement-pots-sle-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-sle-report; output-type=csv;

report measurement-pots-acar-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-acar-report; output-type=csv;

report measurement-pots-cos-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-cos-report; output-type=csv;

report measurement-pots-cot-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pots-cot-report; output-type=csv;

report measurement-ainsvc-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=AIN01; output=ainsvc-report; output-type=csv;

report measurement-sccp-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=AIN01; output=sccp-report; output-type=csv;

report measurement-tcap-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=AIN01; output=tcap-report; output-type=csv;

report measurement-m3ua-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; sgp-id=sg-001; output=m3ua-report; output-type=csv;

report measurement-sua-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; sgp-id=sg-001; output=sua-report; output-type=csv;

report measurement-sctp-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; sctp-assoc-id=assoc-001; output=sctp-report; output-type=csv;

report measurement-isup-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; tgn-id=dallas01; output=isup-report; output-type=csv;

report measurement-audit-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=audit-report; output-type=csv;

report measurement-sia-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=sia-report; output-type=csv;

report measurement-billing-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=billing-report; output-type=csv;

report measurement-em-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=em-report; output-type=csv;

report measurement-dqos-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; aggr-id=AGGR01; output=dqos-report; output-type=csv;
```

```
report measurement-snmp-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; output=snmp-report; output-type=csv;

report measurement-tg-usage-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; tgn-id=dallas01; call-agent-id=CA146; output=tg-report; output-type=csv;

report measurement-tg-usage-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; trkgrp-exchange=RONLVA31GT; trkgrp-name=RONKVACSDS0_LC; call-agent-id=CA146;
output=tg-report; output-type=csv; (this is a new reporting option to gather statistics on
a per Pop basis)

report measurement-anm-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=anm-report; output-type=csv;

report measurement-h323-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=h323-report; output-type=csv;

report measurement-call-tools-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; call-agent-id=CA146; output=call-tools-report; output-type=csv;

report measurement-ain-tools-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=AIN01; output=ain-tools-report; output-type=csv;

report measurement-pct-tools-summary start-time=2007-03-27 10:00:00; end-time=2007-03-27
12:00:00; feature-server-id=PCT01; output=pct-tools-report; output-type=csv;
```

Command Line Tokens

Table 15-6 lists the command line tokens associated with this command and the valid values and purpose of each.

Table 15-6 Command Line Tokens Associated with Measurement Report Summaries

Command Line Token	Description
start-time	A time stamp value in the format of YYYY-MM-DD HH:MM:SS. This value indicates the starting time for the search. This is an optional token. When omitted, it results in the display of the last collected interval.
end-time	A time stamp value in the format of YYYY-MM-DD HH:MM:SS. This value indicates the stopping time for the search. This is an optional token. When omitted, it results in the display of the last collected interval.
interval	This token is optional and is used to specify that a report be generated that contains counter information for the interval currently under collection (current) or all of the collected intervals persisted on disk (all). If this token is used on the command line, it overrides start-time and end-time tokens that are specified. If entered, the corresponding call-agent-id or feature-server-id must be specified. There is no default value for this token. If this token **and** the start-time token are not entered by the user, the last collected interval is reported.
sum	This token indicates whether the resulting report request contains the individual interval reports (N) or a summation of all interval reports into one composite report (Y). The default value for this token is N. This token is not allowed in combination with the trunk group category.

Table 15-6 Command Line Tokens Associated with Measurement Report Summaries (continued)

Command Line Token	Description
output	This token indicates the name of the file to be created and the location where the resulting measurement data is placed. The file name is prepended with the string "Tm_" and placed in the /opt/ems/report directory on the active EMS.
output-type	The format of the output file, which can be in comma-separated value (CSV) or XML format.
display	Allows you to specify the columns of data to present in the resulting report. Only those columns specified are shown in the report. If you enter a value of "%", then a list of all possible column values are displayed, but the report itself is not created.
call-agent-id	The identity of the call agent that collected the measurement data. This is an optional token that defaults to all call agents and is applicable only to the following measurement types: • call-tools • billing • callp • mgcp • isdn • audit • sia • sim • anm • h323 • tg-usage • em
feature-server-id	The identity of the feature server that collected the measurement data. This is an optional token that defaults to all feature servers and is applicable only to the following measurement types: • pct-tools • ain-tools • ainsvc • sccp • tcap • pots-local • pots-misc • pots-sle • pots-acar • pots-cos • pots-cot

Table 15-6 Command Line Tokens Associated with Measurement Report Summaries (continued)

Command Line Token	Description
tgn-id	The trunk group numbers used to report measurement data. This is an optional token that is applicable only to the following measurement types: • tg-usage • isup When used with the trunk measurement type, it results in all trunks within the trunk group being reported.
sgp-id	The signaling gateway process for reporting measurement data. This is an optional token that is applicable only to the following measurement types: • m3ua • sua
sctp-assoc-id	The sctp association ID for reporting measurement data. This is an optional token that is applicable only to the following measurement type: • sctp
aggr-id	The aggregation ID for reporting measurement data. This is an optional token that is applicable only to the following measurement type: • dqos

Reporting Current Interval Counts

The Cisco BTS 10200 system provides a CLI command for querying in-progress partial interval counts of measurement data from the actual source of the data. This mechanism provides the ability for specifying the current collection interval and the particular type and source of data. The start time specified must fall within the current collection interval.

Note This command is not supported for trunk and tg-usage measurement types.

The following are examples of the command line queries for generating reports on the various types of measurements currently being collected from call agents and feature servers on March 27th, 2007, assuming the time is presently 10:05 in the morning:

```
report measurement-isdn-summary call-agent-id=CA146; output=isdn-partial-report;
interval=current; output-type=csv;

report measurement-callp-summary call-agent-id=CA146; output=callp-partial-report;
interval=current; output-type=csv;

report measurement-mgcp-summary call-agent-id=CA146; output=mgcp-partial-report;
interval=current; output-type=csv;

report measurement-sim-summary call-agent-id=CA146; output=sim-partial-report;
interval=current; output-type=csv;

report measurement-pots-local-summary feature-server-id=PCT01;
output=pots-local-partial-report; interval=current; output-type=csv;
```

```
report measurement-pots-misc-summary feature-server-id=PCT01;
output=pots-misc-partial-report; interval=current; output-type=csv;

report measurement-pots-sle-summary feature-server-id=PCT01;
output=pots-sle-partial-report; interval=current; output-type=csv;

report measurement-pots-acar-summary feature-server-id=PCT01;
output=pots-acar-partial-report; interval=current; output-type=csv;

report measurement-pots-cos-summary feature-server-id=PCT01;
output=pots-cos-partial-report; interval=current; output-type=csv;

report measurement-pots-cot-summary feature-server-id=PCT01;
output=pots-cot-partial-report; interval=current; output-type=csv;

report measurement-ainsvc-summary call-agent-id=AIN01; output=ainsvc-partial-report;
interval=current; output-type=csv;

report measurement-sccp-summary call-agent-id=AIN01; output=sccp-partial-report;
interval=current; output-type=csv;

report measurement-tcap-summary call-agent-id=AIN01; output=tcap-partial-report;
interval=current; output-type=csv;

report measurement-audit-summary call-agent-id=CA146; output=audit-partial-report;
interval=current; output-type=csv;

report measurement-sia-summary call-agent-id=CA146; output=sia-partial-report;
interval=current; output-type=csv;

report measurement-billing-summary call-agent-id=CA146; output=billing-partial-report;
interval=current; output-type=csv;

report measurement-em-summary call-agent-id=CA146; output=em-partial-report;
interval=current; output-type=csv;

report measurement-snmp-summary output=snmp-partial-report; interval=current;
output-type=csv;

report measurement-anm-summary call-agent-id=CA146; output=anm-partial-report;
interval=current; output-type=csv;

report measurement-h323-summary call-agent-id=CA146; output=h323-partial-report;
interval=current; output-type=csv;

report measurement-call-tools-summary call-agent-id=CA146;
output=call-tools-partial-report; interval=current; output-type=csv;

report measurement-ain-tools-summary feature-server-id=AIN01;
output=ain-tools-partial-report; interval=current; output-type=csv;

report measurement-pct-tools-summary feature-server-id=PCT01;
output=pct-tools-partial-report; interval=current; output-type=csv;
```

Table 15-7 lists the command line tokens associated with this command and the valid values and purpose of each.

Table 15-7 Command Line Tokens Associated with Reporting Current Interval Counts

Command Line Token	Description
start-time	A time stamp value with the format of YYYY-MM-DD HH:MM:SS. This value indicates the start time for the interval during which a search is made through the EMS database. This is a mandatory token.
output	The name of the file to be created and location to place the resulting measurement data. The file name is prepended with the string "Tm_" and placed in the /opt/ems/report directory on the active EMS.
output-type	The format of the output file—it can be in comma-separated value (CSV) format or XML format.
call-agent-id	The identity of the call agent that collected the measurement data. This is an optional token that defaults to all call agents and is applicable only to the following measurement types: • call-tools • billing • callp • mgcp • isdn • audit • sia • sim • anm • h323 • em

Table 15-7 Command Line Tokens Associated with Reporting Current Interval Counts (continued)

Command Line Token	Description
feature-server-id	The identity of the feature server that collected the measurement data. This is an optional token that defaults to all feature servers and is applicable only to the following measurement types: • ain-tools • pct-tools • ainsvc • sccp • tcap • pots-local • pots-misc • pots-sle • pots-acar • pots-cos • pots-cot
interval	This token is optional and is used to specify that a report be generated that contains counter information for the interval currently under collection (current) or all of the collected intervals currently stored on the disk (all). If this token is used on the command line, it overrides start-time and end-time tokens if they are specified. If entered, the corresponding call-agent-id or feature-server-id must be specified. There is no default value for this token. If this token **and** the start-time token are not entered by the user, the last collected interval is reported.

Clearing Current Interval Counts

The Cisco BTS 10200 system provides a CLI command to clear in-progress partial counts of measurement data at the actual source of the data. This mechanism provides the ability for specifying the particular type and source of data.

Caution ***This is a destructive command that will erase the partial counts for the current interval permanently.*** Use this command with caution.

In the following examples, all of the currently accumulating counters in call agents and feature servers are cleared:

```
clear measurement-isdn-summary call-agent-id=CA146;

clear measurement-callp-summary call-agent-id=CA146;

clear measurement-mgcp-summary call-agent-id=CA146;

clear measurement-sim-summary call-agent-id=CA146;

clear measurement-pots-local-summary feature-server-id=PCT01;
```

```
clear measurement-pots-misc-summary feature-server-id=PCT01;

clear measurement-pots-sle-summary feature-server-id=PCT01;

clear measurement-pots-acar-summary feature-server-id=PCT01;

clear measurement-pots-cos-summary feature-server-id=PCT01;

clear measurement-pots-cot-summary feature-server-id=PCT01;

clear measurement-ainsvc-summary feature-server-id=AIN01;

clear measurement-sccp-summary feature-server-id=AIN01;

clear measurement-sccp-summary feature-server-id=AIN01;

clear measurement-tcap-summary feature-server-id=AIN01;

clear measurement-audit-summary call-agent-id=CA146;

clear measurement-sia-summary call-agent-id=CA146;

clear measurement-billing-summary call-agent-id=CA146;

clear measurement-em-summary call-agent-id=CA146;

clear measurement-snmp-summary

clear measurement-anm-summary call-agent-id=CA146;

clear measurement-h323-summary call-agent-id=CA146;

clear measurement-call-tools-summary call-agent-id=CA146;

clear measurement-ain-tools-summary feature-server-id=AIN01;

clear measurement-pct-tools-summary feature-server-id=PCT01;
```

Table 15-8 is a list of the command line tokens associated with this command and the valid values and purpose of each.

Table 15-8 Command Line Tokens Associated with Clearing Current Interval Counts

Command Line Token	Description
call-agent-id	The identity of the call agent that collected the measurement data. This is an optional token that defaults to all call agents and is applicable only to the following measurement types: • call-tools • billing • callp • mgcp • isdn • audit • sia • sim • anm • H.323 • m3ua • em • sctp
feature-server-id	The identity of the feature server that collected the measurement data. This is an optional token that defaults to all feature servers and is applicable only to the following measurement types: • ain-tools • pct-tools • ainsvc • sccp • tcap • m3ua • sctp • pots-local • pots-misc • pots-sle • pots-acar • pots-cos • pots-cot

Measurements

This section provides detailed information on which counters are maintained within each measurement area. A description of the meaning of each counter is also provided. The name of each counter is an exact ASCII match to the label that is printed within the reports issued by the Cisco BTS 10200. These labels can then be used for automation purposes in testing and retrieving data from the Cisco BTS 10200 through the command line or FTP interfaces.

ISDN Protocol Counters

Table 15-9 identifies the ISDN protocol counters.

Table 15-9 ISDN Protocol Counters

Counter Label	Counter Context
ISDN_SETUP_TX	The number of ISDN setup messages sent from the reporting call agent.
ISDN_SETUP_RX	The number of ISDN setup messages received by the reporting call agent.
ISDN_SETUP_ACK_TX	The number of ISDN setup ACK messages sent from the reporting call agent. This counter is retained for use in a future release.
ISDN_SETUP_ACK_RX	The number of ISDN setup ACK messages received by the reporting call agent. This counter is retained for use in a future release.
ISDN_CALL_PROCEED_TX	The number of ISDN call proceed messages sent from the reporting call agent.
ISDN_CALL_PROCEED_RX	The number of ISDN call proceed messages received by the reporting call agent.
ISDN_ALERTING_TX	The number of ISDN alerting messages sent from the reporting call agent.
ISDN_ALERTING_RX	The number of ISDN alerting messages received by the reporting call agent.
ISDN_PROGRESS_TX	The number of ISDN progress messages sent from the reporting call agent.
ISDN_PROGRESS_RX	The number of ISDN progress messages received by the reporting call agent.
ISDN_CONNECT_TX	The number of ISDN connect messages sent from the reporting call agent.
ISDN_CONNECT_RX	The number of ISDN connect messages received by the reporting call agent.
ISDN_CONNECT_ACK_TX	The number of ISDN connect ACK messages sent from the reporting call agent.
ISDN_CONNECT_ACK_RX	The number of ISDN connect ACK messages received by the reporting call agent.
ISDN_DISCONNECT_TX	The number of ISDN disconnect messages sent from the reporting call agent.
ISDN_DISCONNECT_RX	The number of ISDN disconnect messages received by the reporting call agent.
ISDN_RELEASE_TX	The number of ISDN release messages sent from the reporting call agent.
ISDN_RELEASE_RX	The number of ISDN release messages received by the reporting call agent.
ISDN_RELEASE_COMPLETE_TX	The number of ISDN release complete messages sent from the reporting call agent.
ISDN_RELEASE_COMPLETE_RX	The number of ISDN release complete messages received by the reporting call agent.
ISDN_RESTART_TX	The number of ISDN restart messages sent from the reporting call agent.
ISDN_RESTART_RX	The number of ISDN restart messages received by the reporting call agent.
ISDN_RESTART_ACK_TX	The number of ISDN restart ACK messages sent from the reporting call agent.
ISDN_RESTART_ACK_RX	The number of ISDN restart ACK messages received by the reporting call agent.
ISDN_INFORMATION_TX	The number of ISDN information messages sent from the reporting call agent.

Table 15-9 ISDN Protocol Counters (continued)

Counter Label	Counter Context
ISDN_INFORMATION_RX	The number of ISDN information messages received by the reporting call agent.
ISDN_NOTIFY_TX	The number of ISDN notify messages sent from the reporting call agent.
ISDN_NOTIFY_RX	The number of ISDN notify messages received by the reporting call agent.
ISDN_STATUS_TX	The number of ISDN status messages sent from the reporting call agent.
ISDN_STATUS_RX	The number of ISDN status messages received by the reporting call agent.
ISDN_STATUS_ENQUIRY_TX	The number of ISDN status enquiry messages sent from the reporting call agent.
ISDN_STATUS_ENQUIRY_RX	The number of ISDN status enquiry messages received by the reporting call agent.
ISDN_SRVC_TX	The number of ISDN service messages sent from the reporting call agent.
ISDN_SRVC_RX	The number of ISDN service messages received by the reporting call agent.
ISDN_SRVC_ACK_TX	The number of ISDN service ACK messages sent from the reporting call agent.
ISDN_SRVC_ACK_RX	The number of ISDN service ACK messages received by the reporting call agent.
ISDN_FACILITY_TX	The number of ISDN facility messages sent from the reporting call agent.
ISDN_FACILITY_RX	The number of ISDN facility messages received by the reporting call agent.
ISDN_SUSPEND_TX	The number of ISDN suspend messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SUSPEND_RX	The number of ISDN suspend messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SUSPEND_ACK_TX	The number of ISDN suspend ACK messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SUSPEND_ACK_RX	The number of ISDN suspend ACK messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SUSPEND_REJ_TX	The number of ISDN suspend reject messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SUSPEND_REJ_RX	The number of ISDN suspend reject messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_RESUME_TX	The number of ISDN resume messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_RESUME_RX	The number of ISDN resume messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_RESUME_ACK_TX	The number of ISDN resume ACK messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_RESUME_ACK_RX	The number of ISDN resume ACK messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_RESUME_REJ_TX	The number of ISDN resume reject messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.

Table 15-9 ISDN Protocol Counters (continued)

Counter Label	Counter Context
ISDN_RESUME_REJ_RX	The number of ISDN resume reject messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_USER_INFO_TX	The number of ISDN user information messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_USER_INFO_RX	The number of ISDN user information messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_CONG_CNTL_TX	The number of ISDN congestion control messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_CONG_CNTL_RX	The number of ISDN congestion control messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SEGMENT_TX	The number of ISDN segment messages sent from the reporting call agent. **Note** This counter is applicable only to ETSI PRI.
ISDN_SEGMENT_RX	The number of ISDN segment messages received by the reporting call agent. **Note** This counter is applicable only to ETSI PRI.

Call Processing Counters

Table 15-10 identifies the Call Processing counters and their meanings.

Table 15-10 Call Processing Counters

Counter Label	Counter Context
CALLP_ORIG_ATTMP	The number of originating call attempts of all types on the reporting call agent.
CALLP_TERM_ATTMP	The number of terminating call attempts of all types on the reporting call agent.
CALLP_ORIG_FAIL	The number of originating call attempts of all types that failed on the reporting call agent.
CALLP_TERM_FAIL	The number of terminating call attempts of all types that failed on the reporting call agent.
CALLP_CALL_SUCC	The number of successful originating and terminating call attempts of all types on the reporting call agent.
CALLP_CALL_ABAND	The number of originating call attempts of all types that were abandoned on the reporting call agent.
CALLP_ISDN_ORIG_ATTMP	The number of originating ISDN call attempts on the reporting call agent.
CALLP_ISDN_TERM_ATTMP	The number of ISDN terminating call attempts on the reporting call agent.
CALLP_ISDN_ORIG_FAIL	The number of ISDN originating call attempts that failed on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_ISDN_TERM_FAIL	The number of ISDN terminating call attempts that failed on the reporting call agent.
CALLP_ISDN_CALL_SUCC	The number of successful ISDN originating and terminating call attempts on the reporting call agent.
CALLP_ISDN_CALL_ABAND	The number of ISDN originating call attempts that were abandoned on the reporting call agent.
CALLP_SS7_ORIG_ATTMP	The number of originating SS7 call attempts on the reporting call agent.
CALLP_SS7_TERM_ATTMP	The number of SS7 terminating call attempts on the reporting call agent.
CALLP_SS7_ORIG_FAIL	The number of SS7 originating call attempts that failed on the reporting call agent.
CALLP_SS7_TERM_FAIL	The number of SS7 terminating call attempts that failed on the reporting call agent.
CALLP_SS7_CALL_SUCC	The number of successful SS7 originating and terminating call attempts on the reporting call agent.
CALLP_SS7_CALL_ABAND	The number of SS7 originating call attempts that were abandoned on the reporting call agent.
CALLP_SIP_ORIG_ATTMP	The number of originating SIP call attempts on the reporting call agent.
CALLP_SIP_TERM_ATTMP	The number of SIP terminating call attempts on the reporting call agent.
CALLP_SIP_ORIG_FAIL	The number of SIP originating call attempts that failed on the reporting call agent.
CALLP_SIP_TERM_FAIL	The number of SIP terminating call attempts that failed on the reporting call agent.
CALLP_SIP_CALL_SUCC	The number of successful SIP originating and terminating call attempts on the reporting call agent.
CALLP_SIP_CALL_ABAND	The number of SIP originating call attempts that were abandoned on the reporting call agent.
CALLP_MGCP_ORIG_ATTMP	The number of originating MGCP call attempts on the reporting call agent.
CALLP_MGCP_TERM_ATTMP	The number of MGCP terminating call attempts on the reporting call agent.
CALLP_MGCP_ORIG_FAIL	The number of MGCP originating call attempts that failed on the reporting call agent.
CALLP_MGCP_TERM_FAIL	The number of MGCP terminating call attempts that failed on the reporting call agent.
CALLP_MGCP_CALL_SUCC	The number of successful MGCP originating and terminating call attempts on the reporting call agent.
CALLP_MGCP_CALL_ABAND	The number of MGCP originating call attempts that were abandoned on the reporting call agent.
CALLP_CAS_ORIG_ATTMP	The number of originating CAS call attempts on the reporting call agent.
CALLP_CAS_TERM_ATTMP	The number of CAS terminating call attempts on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_CAS_ORIG_FAIL	The number of CAS originating call attempts that failed on the reporting call agent.
CALLP_CAS_TERM_FAIL	The number of CAS terminating call attempts that failed on the reporting call agent.
CALLP_CAS_CALL_SUCC	The number of successful CAS originating and terminating call attempts on the reporting call agent.
CALLP_CAS_CALL_ABAND	The number of CAS originating call attempts that were abandoned on the reporting call agent.
CALLP_ISDN_SS7_CALL	The number of successfully completed calls from an ISDN originator to an SS7 terminator on the reporting call agent.
CALLP_ISDN_ISDN_CALL	The number of successfully completed calls from an ISDN originator to an ISDN terminator on the reporting call agent.
CALLP_ISDN_SIP_CALL	The number of successfully completed calls from an ISDN originator to an SIP terminator on the reporting call agent.
CALLP_ISDN_MGCP_CALL	The number of successfully completed calls from an ISDN originator to an MGCP terminator on the reporting call agent.
CALLP_ISDN_CAS_CALL	The number of successfully completed calls from an ISDN originator to an CAS terminator on the reporting call agent.
CALLP_SS7_SS7_CALL	The number of successfully completed calls from an SS7 originator to an SS7 terminator on the reporting call agent.
CALLP_SS7_ISDN_CALL	The number of successfully completed calls from an SS7 originator to an ISDN terminator on the reporting call agent.
CALLP_SS7_SIP_CALL	The number of successfully completed calls from an SS7 originator to an SIP terminator on the reporting call agent.
CALLP_SS7_MGCP_CALL	The number of successfully completed calls from an SS7 originator to an MGCP terminator on the reporting call agent.
CALLP_SS7_CAS_CALL	The number of successfully completed calls from an SS7 originator to an CAS terminator on the reporting call agent.
CALLP_SIP_SS7_CALL	The number of successfully completed calls from a SIP originator to an SS7 terminator on the reporting call agent.
CALLP_SIP_ISDN_CALL	The number of successfully completed calls from a SIP originator to an ISDN terminator on the reporting call agent.
CALLP_SIP_SIP_CALL	The number of successfully completed calls from a SIP originator to an SIP terminator on the reporting call agent.
CALLP_SIP_MGCP_CALL	The number of successfully completed calls from a SIP originator to an MGCP terminator on the reporting call agent.
CALLP_SIP_CAS_CALL	The number of successfully completed calls from a SIP originator to an CAS terminator on the reporting call agent.
CALLP_MGCP_SS7_CALL	The number of successfully completed calls from an MGCP originator to an SS7 terminator on the reporting call agent.
CALLP_MGCP_ISDN_CALL	The number of successfully completed calls from an MGCP originator to an ISDN terminator on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_MGCP_SIP_CALL	The number of successfully completed calls from an MGCP originator to an SIP terminator on the reporting call agent.
CALLP_MGCP_MGCP_CALL	The number of successfully completed calls from an MGCP originator to an MGCP terminator on the reporting call agent.
CALLP_MGCP_CAS_CALL	The number of successfully completed calls from an MGCP originator to an CAS terminator on the reporting call agent.
CALLP_CAS_SS7_CALL	The number of successfully completed calls from a CAS originator to an SS7 terminator on the reporting call agent.
CALLP_CAS_ISDN_CALL	The number of successfully completed calls from a CAS originator to an ISDN terminator on the reporting call agent.
CALLP_CAS_SIP_CALL	The number of successfully completed calls from a CAS originator to an SIP terminator on the reporting call agent.
CALLP_CAS_MGCP_CALL	The number of successfully completed calls from a CAS originator to an MGCP terminator on the reporting call agent.
CALLP_CAS_CAS_CALL	The number of successfully completed calls from a CAS originator to a CAS terminator on the reporting call agent.
CALLP_INTERLA_ATTMP	The number of interLATA call attempts on the reporting call agent.
CALLP_INTERLA_FAIL	The number of interLATA call attempts that failed on the reporting call agent.
CALLP_INTERLA_SUCC	The number of interLATA call attempts that completed successfully on the reporting call agent.
CALLP_INTERLA_ABAND	The number of interLATA call origination attempts that were abandoned on the reporting call agent.
CALLP_INTRALA_ATTMP	The number of intraLATA call attempts on the reporting call agent.
CALLP_INTRALA_FAIL	The number of intraLATA call attempts that failed on the reporting call agent.
CALLP_INTRALA_SUCC	The number of intraLATA call attempts that completed successfully on the reporting call agent.
CALLP_INTRALA_ABAND	The number of intraLATA call origination attempts that were abandoned on the reporting call agent.
CALLP_INTL_ATTMP	The number of international call attempts on the reporting call agent.
CALLP_INTL_FAIL	The number of international call attempts that failed on the reporting call agent.
CALLP_INTL_SUCC	The number of international call attempts that completed successfully on the reporting call agent.
CALLP_INTL_ABAND	The number of international call origination attempts that were abandoned on the reporting call agent.
CALLP_EMGNCY_ATTMP	The number of emergency call attempts on the reporting call agent.
CALLP_EMGNCY_FAIL	The number of emergency call attempts that failed on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_EMGNCY_CALL_SUCC	The number of emergency call attempts that completed successfully on the reporting call agent.
CALLP_EMGNCY_CALL_ABAND	The number of emergency call origination attempts that were abandoned on the reporting call agent.
CALLP_LOCAL_ATTMP	The number of local call attempts on the reporting call agent.
CALLP_LOCAL_FAIL	The number of local call attempts that failed on the reporting call agent.
CALLP_LOCAL_SUCC	The number of local call attempts that completed successfully on the reporting call agent.
CALLP_LOCAL_ABAND	The number of local call origination attempts that were abandoned on the reporting call agent.
CALLP_TOLL_FREE_ATTMP	The number of toll free call attempts on the reporting call agent.
CALLP_TOLL_FREE_FAIL	The number of toll free call attempts that failed on the reporting call agent.
CALLP_TOLL_FREE_SUCC	The number of toll free call attempts that completed successfully on the reporting call agent.
CALLP_TOLL_FREE_ABAND	The number of toll free call origination attempts that were abandoned on the reporting call agent.
CALLP_H323_ORIG_ATTMP	The number of originating H.323 call attempts on the reporting call agent.
CALLP_H323_TERM_ATTMP	The number of terminating H.323 call attempts on the reporting call agent.
CALLP_H323_ORIG_FAIL	The number of originating H.323 call attempts that failed on the reporting call agent.
CALLP_H323_TERM_FAIL	The number of terminating H.323 call attempts that failed on the reporting call agent.
CALLP_H323_CALL_SUCC	The number of originating and terminating H.323 call attempts that completed successfully on the reporting call agent.
CALLP_H323_CALL_ABAND	The number of terminating and originating H.323 call attempts that were abandoned on the reporting call agent.
CALLP_ISDN_H323_CALL	The total number of successfully completed calls from an ISDN originator to an H.323 terminator on the reporting call agent.
CALLP_SS7_H323_CALL	The total number of successfully completed calls from an SS7 originator to an H.323 terminator on the reporting call agent.
CALLP_SIP_H323_CALL	The total number of successfully completed calls from a SIP originator to an H.323 terminator on the reporting call agent.
CALLP_MGCP_H323_CALL	The total number of successfully completed calls from an MGCP originator to an H.323 terminator on the reporting call agent.
CALLP_CAS_H323_CALL	The total number of successfully completed calls from a CAS originator to an H.323 terminator on the reporting call agent.
CALLP_H323_SIP_CALL	The total number of successfully completed calls from an H.323 originator to a SIP terminator on the reporting call agent.
CALLP_H323_ISDN_CALL	The total number of successfully completed calls from an H.323 originator to an ISDN terminator on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_H323_SS7_CALL	The total number of successfully completed calls from an H.323 originator to an SS7 terminator on the reporting call agent.
CALLP_H323_MGCP_CALL	The total number of successfully completed calls from an H.323 originator to an MGCP terminator on the reporting call agent.
CALLP_H323_CAS_CALL	The total number of successfully completed calls from an H.323 originator to a CAS terminator on the reporting call agent.
CALLP_H323_H323_CALL	The total number of successfully completed calls from an H.323 originator to an H.323 terminator on the reporting call agent.
CALLP_NAS_AUTH_SUCC	The total number of successful NAS authentication requests on the reporting call agent.
CALLP_NAS_AUTH_FAIL	The total number of failed NAS authentication requests on the reporting call agent.
CALLP_NAS_OP_FAIL	The total number of operation failures that occurred on the reporting call agent–typically indicative of a modem failure.
CALLP_NAS_ISP_PORT_LIMIT	The total number of NAS calls that failed on the reporting call agent due to the port limit of a modem being exceeded.
CALLP_NAS_NO_MODEMS	The total number of NAS calls that failed on the reporting call agent due to the unavailability of a modem.
CALLP_NAS_CLG_UNACC	The total number of NAS calls that failed on the reporting call agent due to the calling party number being blocked.
CALLP_NAS_CLD_UNACC	The total number of NAS calls that failed on the reporting call agent due to the called party number being blocked.
CALLP_NAS_USER_REQUEST	The total number of user requests (Reason Code 801) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_LOST_CARRIER	The total number of lost carrier hits (Reason Code 802) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_LOST_SERVICE	The total number of lost service hits (Reason Code 803) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_IDLE_TIMEOUT	The total number of idle timeouts (Reason Code 804) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_SESSION_TIMEOUT	The total number of session timeouts (Reason Code 805) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_ADMIN_RESET	The total number of adminstrator resets (Reason Code 806) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_ADMIN_REBOOT	The total number of adminstrator reboots (Reason Code 807) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_PORT_ERROR	The total number of port errors (Reason Code 808) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_NAS_ERROR	The total number of NAS errors (Reason Code 809) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_NAS_REQUEST	The total number of NAS requests (Reason Code 810) that are received in the DLCX messages on the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_NAS_NAS_REBOOT	The total number of NAS reboots (Reason Code 811) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_PORT_UNNEEDED	The total number of port unneeded hits (Reason Code 812) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_PORT_PREEMPTED	The total number of port preempted hits (Reason Code 813) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_PORT_SUSPENDED	The total number of port suspended hits (Reason Code 814) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_SERVICE_UNAVAIL	The total number of service unavailable hits (Reason Code 815) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_CALLBACK	The total number of NAS callbacks (Reason Code 816) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_USER_ERROR	The total number of user errors (Reason Code 817) that are received in the DLCX messages on the reporting call agent.
CALLP_NAS_HOST_REQUEST	The total number of host requests (Reason Code 818) that are received in the DLCX messages on the reporting call agent.
CALLP_IVR_NETWORK_REQ	The total number of requests for network based IVR service on the reporting call agent.
CALLP_IVR_NATIVE_REQ	The total number of requests for native IVR service on the reporting call agent.
CALLP_IVR_RESOURCE_FAIL	The total number of IVR sessions that could not be established on the reporting call agent.
CALLP_TOTAL_TDISC_ORIG_ATTMP	The total number of origination attempts by subscribers that are marked as temporarily disconnected, detected by the reporting call agent.
CALLP_ NLB_TEST_SUCC	The total number of successful network loop back tests completed by the reporting call agent.
CALLP_ NLB_TEST_FAIL	The total number of failed network loop back tests completed by the reporting call agent. This counter includes both call setup failures and resource failures. These are test calls abnormally released by the call agent due to reasons such as resource priorities.
CALLP_NCT_TEST_SUCC	The total number of successful network continuity tests completed by the reporting call agent.
CALLP_ NCT_TEST_FAIL	The total number of failed network continuity tests completed by the reporting call agent. This counter includes both call setup failures and resource failures. These are test calls abnormally released by the call agent due to reasons such as resource priorities.
CALLP_LB_TEST_SUCC	The total number of successful TDM loop back tests (108) completed by the reporting call agent.
CALLP_TEST_ROUTE_SUCC	The total number of successful TDM loop back tests (108) with DN dialed out in outgoing message completed by the reporting call agent.

Table 15-10 Call Processing Counters (continued)

Counter Label	Counter Context
CALLP_T38_FAX_MEDIA_SETUP_SUCC	This counter is incremented when the T.38 media connection is established successfully between the endpoints for T.38 fax transmission.
CALLP_T38_FAX_MEDIA_SETUP_FAIL	This counter is incremented when a T.38 media connection is not established successfully between the endpoints for T.38 fax transmission.

MGCP Adapter Counters

Table 15-11 identifies the MGCP Adapter counters.

Table 15-11 MGCP Adapter Counters

Counter Label	Counter Context
MGCP_DECODE_ERROR	The number of MGCP messages received that failed decoding on the reporting call agent.
MGCP_ENCODE_ERROR	The number of MGCP messages to be sent that failed encoding on the reporting call agent.
MGCP_UNREACHABLE	The number of MGCP messages sent from the reporting call agent that failed due to the target gateway being unreachable.
MGCP_SEND_FAILED	The number of MGCP messages sent from the reporting call agent that failed while being sent to the target gateway.
MGCP_CRCX_ACK_RX	The number of MGCP CRCX acknowledgement messages received by the reporting call agent.
MGCP_CRCX_NACK_RX	The number of MGCP CRCX nonacknowledgement messages received by the reporting call agent.
MGCP_CRCX_TX	The number of MGCP CRCX messages sent by the reporting call agent.
MGCP_MDCX_ACK_RX	The number of MGCP MDCX acknowledgement messages received by the reporting call agent.
MGCP_MDCX_NACK_RX	The number of MGCP MDCX nonacknowledgement messages received by the reporting call agent.
MGCP_MDCX_TX	The number of MGCP MDCX messages sent by the reporting call agent.
MGCP_DLCX_RX	The number of MGCP DLCX messages received from gateways by the reporting call agent.
MGCP_DLCX_TX	The number of MGCP DLCX messages s.ent by the reporting call agent
MGCP_DLCX_ACK_RX	The number of MGCP DLCX acknowledgement messages received by the reporting call agent.
MGCP_DLCX_NACK_RX	The number of MGCP DLCX nonacknowledgement messages received by the reporting call agent.
MGCP_RQNT_ACK_RX	The number of MGCP RQNT acknowledgement messages received by the reporting call agent.
MGCP_RQNT_NACK_RX	The number of MGCP RQNT nonacknowledgement messages received by the reporting call agent.
MGCP_RQNT_TX	The number of MGCP RQNT messages sent by the reporting call agent.
MGCP_AUEP_ACK_RX	The number of MGCP AUEP acknowledgement messages received by the reporting call agent.

Table 15-11 MGCP Adapter Counters

Counter Label	Counter Context
MGCP_AUEP_NACK_RX	The number of MGCP AUEP nonacknowledgement messages received by the reporting call agent.
MGCP_AUEP_TX	The number of MGCP AUEP messages sent by the reporting call agent.
MGCP_NTFY_RX	The number of MGCP NTFY messages received from gateways by the reporting call agent.
MGCP_RSIP_RX	The number of MGCP RSIP messages received from gateways by the reporting call agent.
MGCP_RSIP_ACK_TX	The number of MGCP RSIP acknowledgement messages sent by the reporting call agent.
MGCP_AUCX_TX	The number of AUCX (audit connection) messages that were sent by the reporting call agent.
MGCP_AUCX_ACK_RX	The number of AUCX ACK (audit connection acknowledgement) messages that were received by the reporting call agent.
MGCP_AUCX_NACK_RX	The number of AUCX NACK (audit connection nonacknowlegement) messages that were received by the reporting call agent.

Session Initiation Protocol Counters

Table 15-12 identifies the Session Initiation Protocol counters. These counters are common to several reporting types including SIM, AIN-SVC, POTS-MISC, and SIA.

Table 15-12 Session Initiation Protocol Counters

Counter Label	Counter Context
SIS_TOTAL_INCOM_MSG	The number of SIP messages the reporting call agent or feature server attempted to receive.
SIS_TOTAL_SUCC_INCOM_MSG	The number of SIP messages the reporting call agent or feature server successfully received.
SIS_TOTAL_OUTG_MSG_ATTMP	The number of SIP messages the reporting call agent or feature server attempted to send.
SIS_TOTAL_SUCC_OUTG_MSG	The number of SIP messages the reporting call agent or feature server successfully sent.
SIS_REQ_RETRAN_RX	The number of SIP request retransmission messages the reporting call agent or feature server received.
SIS_REQ_RETRAN_TX	The number of SIP request retransmission messages the reporting call agent or feature server sent.
SIS_RSP_RETRAN_RX	The number of SIP response retransmission messages the reporting call agent or feature server received.
SIS_RSP_RETRAN_TX	The number of SIP response retransmission messages the reporting call agent or feature server sent.
SIS_T1_TIMER_EXPIRED	The number of SIP T1 timer expirations that occurred on the reporting call agent or feature server received over the collection interval.
SIS_T2_TIMER_REACHED	The number of SIP T2 timer expirations that occurred on the reporting call agent or feature server received over the collection interval.
SIS_INVITE_RX	The number of SIP invite messages the reporting call agent or feature server received.
SIS_INVITE_TX	The number of SIP invite messages the reporting call agent or feature server sent.

Table 15-12 Session Initiation Protocol Counters (continued)

Counter Label	Counter Context
SIS_CANCEL_RX	The number of SIP cancel messages the reporting call agent or feature server received.
SIS_CANCEL_TX	The number of SIP cancel messages the reporting call agent or feature server sent.
SIS_BYE_RX	The number of SIP bye messages the reporting call agent or feature server received.
SIS_BYE_TX	The number of SIP bye messages the reporting call agent or feature server sent.
SIS_ACK_RX	The number of SIP acknowledgement messages the reporting call agent or feature server received.
SIS_ACK_TX	The number of SIP acknowledgement messages the reporting call agent or feature server sent.
SIS_OPTIONS_RX	The number of SIP options messages the reporting call agent or feature server received.
SIS_OPTIONS_TX	The number of SIP options messages the reporting call agent or feature server sent.
SIS_REGISTER_RX	The number of SIP register messages the reporting call agent or feature server received.
SIS_REGISTER_TX	The number of SIP register messages the reporting call agent or feature server sent.
SIS_INFO_RX	The number of SIP informational messages the reporting call agent or feature server received.
SIS_INFO_TX	The number of SIP informational messages the reporting call agent or feature server sent.
SIS_NOTIFY_RX	The number of SIP notify messages the reporting call agent or feature server received.
SIS_NOTIFY_TX	The number of SIP notify messages the reporting call agent or feature server sent.
SIS_100_RX	The number of 100 class (trying) messages the reporting call agent or feature server received.
SIS_100_TX	The number of 100 class (trying) messages the reporting call agent or feature server sent.
SIS_18x_RX	The number of 18x class (informational) messages the reporting call agent or feature server received.
SIS_18x_TX	The number of 18x class (informational) messages the reporting call agent or feature server sent.
SIS_200_RX	The number of 200 class (success) messages the reporting call agent or feature server received.
SIS_200_TX	The number of 200 class (success) messages the reporting call agent or feature server sent.
SIS_3xx_RX	The number of 3xx class (redirection) messages the reporting call agent or feature server received.
SIS_3xx_TX	The number of 3xx class (redirection) messages the reporting call agent or feature server sent.
SIS_4xx_RX	The number of 4xx class (request failures) messages the reporting call agent or feature server received.
SIS_4xx_TX	The number of 4xx class (request failures) messages the reporting call agent or feature server sent.

Table 15-12 Session Initiation Protocol Counters (continued)

Counter Label	Counter Context
SIS_5xx_RX	The number of 5xx class (server failures) messages the reporting call agent or feature server received.
SIS_5xx_TX	The number of 5xx class (server failures) messages the reporting call agent or feature server sent.
SIS_6xx_RX	The number of 6xx class (global failures) messages the reporting call agent or feature server received.
SIS_6xx_TX	The number of 6xx class (global failures) messages the reporting call agent or feature server sent.
SIS_7xx_RX	The number of 7xx class (reserved) messages the reporting call agent or feature server received.
SIS_7xx_TX	The number of 7xx class (reserved) messages the reporting call agent or feature server sent.
SIS_PROV_RSP_RETRAN_RX	The number of SIP provisioning response retransmission messages the reporting call agent or feature server received.
SIS_PROV_RSP_RETRAN_TX	The number of SIP provisioning response retransmission messages the reporting call agent or feature server sent.
SIS_PRACK_RX	The number of SIP PRACK messages the reporting call agent or feature server received.
SIS_PRACK_TX	The number of SIP PRACK messages the reporting call agent or feature server sent.
SIS_SUBSCRIBE_RX	The number of SIP subscribe messages the reporting call agent or feature server received.
SIS_SUBSCRIBE_TX	The number of SIP subscribe messages the reporting call agent or feature server sent.
SIS_REFER_RX	The number of SIP refer messages the reporting call agent or feature server received.
SIS_REFER_TX	The number of SIP refer messages the reporting call agent or feature server sent.
SIS_REFER_W_REPLACES_RX	The number of SIP refer with replaces messages the reporting call agent or feature server received.
SIS_INVITE_REPLACES_TX	The number of SIP invite replaces messages the reporting call agent or feature server sent.
SIS_INVITE_REPLACES_RX	The number of SIP invite replaces messages the reporting call agent or feature server received.
SIS_REL100_RX	The number of REL100 class (trying) messages the reporting call agent or feature server received.
SIS_REL100_TX	The number of REL100 class (trying) messages the reporting call agent or feature server sent.
SIS_UNSUPPORTED_RX	The number of unsupported SIP messages the reporting call agent or feature server received.
SIS_UPDATE_RX	The number of SIP update messages the reporting call agent or feature server received.
SIS_UPDATE_TX	The number of SIP update messages the reporting call agent or feature server sent.

Cisco BTS 10200 Status

The Cisco BTS 10200 status (BTSSTAT) software utility provides status information for the entire Cisco BTS 10200 system. It can run on any Cisco BTS 10200 host and report the status of all the network elements in the Cisco BTS 10200 system, including those not on the same host. BTSSTAT is designed to be fast and secure.

The operator can execute the **btsstat** command from the UNIX shell on any host of a Cisco BTS 10200 system. The operator can be any valid UNIX user.

The output of BTSSTAT includes the network element id, side, host name, version, replication status, and redundancy status of all Cisco BTS 10200 network elements. All of the results appear in one screen. A sample of the output is shown in Table 15-13.

Table 15-13 Sample BTSSTAT Output

```
prica16# btsstat
 ------------------------------------------------------------------------
| ID-SIDE (HOST)   | CA146-A(prica16)            | CA146-B(secca16)            |
| VERSION          | 900-05.00.00.I06            | 900-05.00.00.I06            |
| RED, REPL STATE | STANDBY, Replicating        | ACTIVE, Replicating         |
|-----------------+-----------------------------+-----------------------------|
| ID-SIDE (HOST)   | FSAIN205-A(prica16)         | FSAIN205-B(secca16)         |
| VERSION          | 900-05.00.00.I06            | 900-05.00.00.I06            |
| RED, REPL STATE | ACTIVE, Replicating         | STANDBY, Replicating        |
|-----------------+-----------------------------+-----------------------------|
| ID-SIDE (HOST)   | FSPTC235-A(prica16)         | FSPTC-B(secca16)            |
| VERSION          | 900-05.00.00.I06            |                             |
| RED, REPL STATE | ACTIVE, Not Replicating     | No response/OOS             |
|-----------------+-----------------------------+-----------------------------|
| ID-SIDE (HOST)   | EM01-A(priems16)            | EM01-B(secems16)            |
| VERSION          | 900-05.00.00.I06            | 900-05.00.00.I06            |
| RED, REPL STATE | ACTIVE, Replicating         | STANDBY, Replicating        |
|-----------------+-----------------------------+-----------------------------|
| ID-SIDE (HOST)   | BDMS01-A(priems16)          | BDMS01-B(secems16)          |
| VERSION          | 900-05.00.00.I06            | 900-05.00.00.I06            |
| RED, REPL STATE | ACTIVE, Replicating         | STANDBY, Replicating        |
 ------------------------------------------------------------------------
prica16#
```

By default, BTSSTAT relies on /etc/opticall.cfg to find the host name for each Cisco BTS 10200 network element, and uses the default TCP port numbers of the Platform Application Services (PAS) server (Table 15-14) on each side of the network element to establish an SSL connection to it and to obtain information.

Table 15-14 Default TCP Port Number of PAS Server

Application	Default Port Number
CA	16001
FSAIN	16002
FSPTC	16003
EMS	16004
BDMS	16005

Note Both sides of one Cisco BTS 10200 network element use the same port.

You can run BTSSTAT from a host that is not a Cisco BTS 10200, provided that the host can establish an SSL connection to the target Cisco BTS 10200 host. In this case, the Cisco BTS 10200 hosts should be specified in a configuration file. Users can specify a configuration file with the -f option as follows:

```
btsstat -f my_cfg_file
```

The user-provided configuration file must contain the tokens in Table 15-15 along with the values of the corresponding host names. BTSSTAT ignores all other lines in the file.

Table 15-15 BTSSTAT Configuration File Format

Element	Setting
CA_SIDE_A_HN = CA_SIDE_B_HN =	prica11 secca11
FSAIN_SIDE_A_HN = FSAIN_SIDE_B_HN =	prica11 secca11
FSPTC_SIDE_A_HN = FSPTC_SIDE_B_HN =	prica11 secca11
EMS_SIDE_A_HN = EMS_SIDE_B_HN =	priems11 secems11
BDMS_SIDE_A_HN = BDMS_SIDE_B_HN =	priems11 secems11

You can use a command-line argument to specify nondefault port numbers to status for any of the Cisco BTS 10200 network elements. The command-line options in Table 15-16 are for specifying the port numbers.

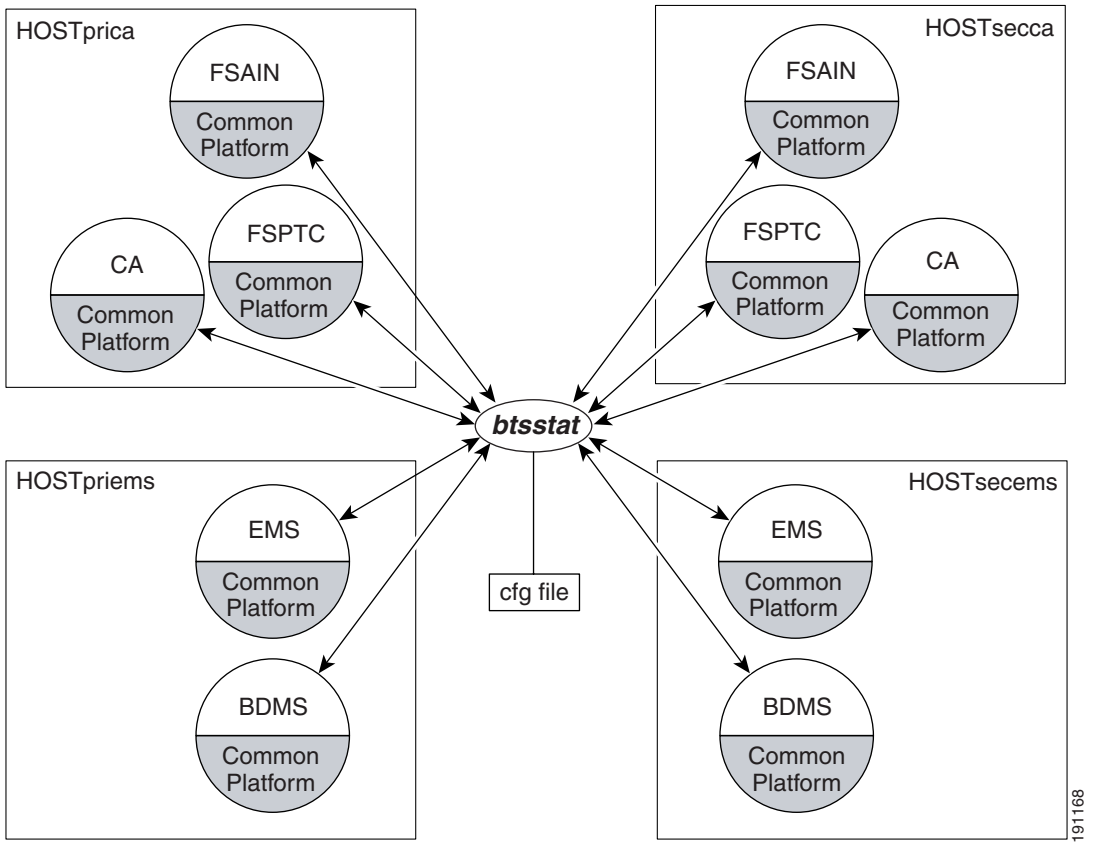
Table 15-16 BTSSTAT Command-line Options for Specifying Port Numbers for Statusing

Command Line Option	Description	Example
-caport <num>	Specify the port number for CA	btsstat -caport 16007
-fsainport <num>	Specify the port number for FSAIN	btsstat -fsainport 16008
-fsptcport <num>	Specify the port number for FSPTC	btsstat -fsptcport 16009
-emsport <num>	Specify the port number for EMS	btsstat -emsport 16010
-bdmsport <num>	Specify the port number for BDMS	btsstat -bdmsport 16011

System Context for BTSSTAT

BTSSTAT queries all the network elements in the same Cisco BTS 10200 system for the status information shown in Figure 15-1. Additionally, Figure 15-1 illustrates the interrelated conditions or context for which the **btsstat** command provides status. BTSSTAT can run on any of these Cisco BTS 10200 hosts, or it can run on a separate host.

Figure 15-1 BTSSTAT System Context



Prerequisites

The BTSSTAT software utility needs Apache xerces-c library Version 2.6.0 or higher to parse and serialize the XML message. This shared library must be present in the host in order for you to run BTSSTAT.

Installing

Use the following procedures to install the BTSSTAT software utility on a Cisco BTS 10200 host and on a host that is not a Cisco BTS 10200.

Installation on a Cisco BTS 10200 Host

BTSSTAT is part of BTSTOOLS package. This package is installed automatically when you install Cisco BTS 10200. The tool is in the /opt/bts/bin directory after installation.

No specific installation/upgrade/fallback procedure is required for this tool.

Installation on a Host That Is Not a Cisco BTS 10200

To install BTSSTAT on a host that is not a Cisco BTS 10200:

Step 1 Make sure that the host is Solaris-SPARC based and that the SSL connection from the host to the target Cisco BTS 10200 system is allowed.

Step 2 Obtain the BTSSTAT executable file and the XML parser library.

On an installed Cisco BTS 10200 system, the two files are located at /opt/bts/bin/btsstat and /opt/BTSlib/lib/libxerces-c.so.26.

Step 3 Transfer the two files into the host that is not a Cisco BTS 10200.

Step 4 Make sure that the BTSSTAT file has the correct permissions and that the library file libxerces-c.so.26 is in $LD_LIBRARY_PATH.

Step 5 Provide your own configuration file (see Table 15-15).

Step 6 Now the **btsstat** command can be run as
btsstat -f cfg_file

For upgrade, the two files can be simply overwritten.

For fallback, the two files can be simply replaced by the previous version.

Call Tracer (CTRAC)

The Cisco BTS 10200 call tracer (CTRAC) feature provides a mechanism that uniquely marks each Cisco BTS 10200 system call to provide a system call trace troubleshooting capability.

The CTRAC feature provides an easy means to filter out trace log lines that correspond to a specific basic or feature call. The filtering is enabled by a unique CTRAC-ID set unconditionally for every call attempt (at the earliest point in time in call processing) and provides a copy of it to all call-processing modules in the Cisco BTS 10200 (across platforms). The CTRAC-ID is used for logging seamlessly into per-call related trace lines corresponding to the call.

Because every per-call related trace log line has a CTRAC-ID, a user can use UNIX **grep** or a similar command to filter out the lines of interest using the CTRAC-ID.

Restrictions and Limitations

Note This feature is available only to users with both CLI and root (UNIX) access.

Due to implementation limitations, it is possible that some per-call related trace logs may not have CTRAC-IDs. Such occurrences however are limited in number.

The CCB shared memory used by various modules is affected. The CCB structure to is expanded include CTRAC-ID.

Operating

The CTRAC feature is the key enabling feature for the end user interested in troubleshooting or debugging calls by viewing the Cisco BTS 10200 trace logs. CTRAC enables the system user to collect all Cisco BTS 10200 trace logs pertaining to a single call. Please refer to the following sections for examples of using the Cisco BTS 10200 CTRAC feature:

- Isolating Calls Based on Billing Record, page 15-89
- Isolating Calls Based on a Given Originating End Point, page 15-89
- Isolating Calls Based on a Given Terminating End Point, page 15-90
- Isolating Calls Which Show Internal Symptoms of Problems, page 15-90

Isolating Calls Based on Billing Record

To isolate calls based on the billing record, take the following steps:

Step 1 For a given call of interest, note (through CLI) the value of the CTRAC-ID billing record parameter. This value is the CTRAC-ID for the call. For this example, assume that it is M0000001. The CTRAC billing-cdr parameter is CTRACID. It can be obtained by using the CLI **report billing-record** command.

Step 2 For each call-processing platform of interest (CA, FSPTC, FSAIN, BDMS), go to the directory where the Cisco BTS 10200 trace logs are stored (by default this is the /opt/OptiCall/<platform-instance-name>/bin/logs directory).

Note If the trace log files are zipped by the platform, you have to copy the zipped files to a separate directory and perform the necessary operations to unzip the file in the separate directory.

```
$ cd /opt/OptiCall/<platform-instance-name>/bin/logs
```

Where platform-instance-name could be CA146, FSPTC235, or the name of some other platform installed in your system.

Step 3 In the directory where the Cisco BTS 10200 trace logs are available, use the UNIX **grep** command to filter out the trace logs corresponding to the selected CTRAC-ID.

```
$  grep "M0000001" *.log > CTRAC-M0000001.txt
```

Step 4 View the CTRAC-M0000001.txt file with a text editor to browse the trace log file lines corresponding to the call.

Isolating Calls Based on a Given Originating End Point

To isolate calls based on a given originating end point, take the following steps:

Step 1 Go to the desired target platform log directory.

```
$ cd /opt/OptiCall/<platform-instance-name>/bin/logs
```

Step 2 Use the UNIX **grep** command to scan for a line of specified format to filter out the term-id/idx to the CTRAC-ID correlation log line.

```
$ grep "OHALF_CTRAC_MAP" | grep "my-term-id-here"
```

The information in the resultant line correlates with the CTRAC-ID for all calls that originated from the specified endpoint.

Step 3 Use the CTRAC-ID to filter out the trace log lines from the Cisco BTS 10200 logs.

Isolating Calls Based on a Given Terminating End Point

To isolate calls based on a given terminating end point, take the following steps:

Step 1 Go to the desired target platform log directory.

```
$ cd /opt/OptiCall/<platform-instance-name>/bin/logs
```

Step 2 Use the UNIX **grep** command to scan for line of specified format to filter out the term-id / idx to CTRAC-ID correlation log line.

```
$ grep "THALF_CTRAC_MAP" | grep "<my-term-id-here>"
```

The information in the resultant line correlates with the CTRAC-ID for all calls that terminated at the specified endpoint.

Step 3 Use the CTRAC-ID to filter out the trace log lines from the Cisco BTS 10200 logs.

Isolating Calls Which Show Internal Symptoms of Problems

To use error and warn messages isolate calls which show internal symptoms of problems, take the following steps:

Step 1 Go to the desired target platform log directory.

```
$ cd /opt/OptiCall/<platform-instance-name>/bin/logs
```

Step 2 Use the UNIX **grep** command to scan for lines with error or warn messages.

```
$  grep "ERROR" *.log
```

Step 3 If you find a nonzero CTRAC-ID present in the appropriate column in the trace log, it means that the error occurred while a call was being processed. Note the CTRAC-ID.

Step 4 Use the CTRAC-ID to filter out the trace log lines from the Cisco BTS 10200 logs.

Billing Fields

The billing record contains a new parameter that contains the CTRAC-ID related to the feature described in this document. The CTRAC billing CDR parameter is named CTRACID.

Troubleshooting

The Cisco BTS 10200 CTRAC feature is intended as a troubleshooting enabler. No specific troubleshooting steps are required other than the use of **grep** to filter out the trace lines corresponding to a CTRAC-ID.

Tabular Display of Events and Alarms

The Cisco BTS 10200 tabular display of events and alarms feature enables the Cisco BTS 10200 to display current alarms, alarm history and event history in tabular form. The underlying CPI layer modification will easily facilitate other commands to display their data in tabular form.

The Cisco BTS 10200 tabular display of events and alarms feature enables the Cisco BTS 10200 to display current alarms, alarm history and event history in tabular form. The output of the **show alarm** and the **show alarm-log** commands will provide a tabular output consisting of one alarm per row. Each of the reported fields will be columns in the tabular output. This will make the outputs much more conducive to capture and printing.

CPI layer changes will facilitate the tabular display of alarms. The CPI layer changes will allow the tabular display of other data through derived request managers.

Operating

There are three new CLI commands related to the Cisco BTS 10200 tabular display of events and alarms feature correlating to the following:

- Show current alarms in tabular format
- Show alarms history in tabular format
- Show events history in tabular format

CLI Commands

The following are show tab CLI commands and their description:

- **show tab-alarm**—Shows current alarms in tabular format
- **show tab-alarm-hist**—Shows alarms history in tabular format
- **show tab-event-hist**—Shows event history in tabular format

An example of the output of executing of one the three commands follows:

```
CLI> show tab-alarm

ID            TYPE          NUMBER   Severity   TIMESTAMP              COMPONENT-ID
12331874656   SIGNALING     68       MAJOR      2005-11-20 11:00:00    testing@mgw_id
12331874657   MAINTENANCE   3        MAJOR      2005-11-20 11:01:00    testing
12331874658   OSS           9        MINOR      2005-11-20 11:02:45    unixserver
12331874659   OSS           9        MINOR      2005-11-20 11:02:45    skittles
```

Prior to Manual Switchover Switch Integrity Diagnostic Utility

This section describes the prior to manual switchover switch integrity diagnostic utility feature that provides the switchover target system health information. For costly traffic outages to be avoided, the switchover decision must be made based on the system health information. The diagnostic script utility automates the manual procedures that are used to collect, check, and verify system health information.

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature provides the system health information for the switchover target so that Cisco BTS 10200 customers can decide if they want to perform manual switchover.

The customer service provider operation organizations often periodically perform switch overs during maintenance windows (after midnight and early in the morning). Doing so ensures that the mate system is functional and cleans up any abnormalities that might have accumulated in the current active/primary system (for example, memory leaks, hung processes).

In order to facilitate the switchover operational practice, the Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature provides a diagnostic tool/utility that allows the operator to verify the operational status of the standby system/platform and decide whether it is in a ready state for a switchover.

Note See the *Cisco BTS 10200 Softswitch Network and Subscriber Feature Descriptions* guide for a complete list of subscriber features supported by the Cisco BTS 10200.

Application Status Check

Before performing a manual switchover, you need to ensure that the switchover target is in the standby state. The primary target of the utility is CA application, the utility also checks FS applications to verify that all three applications (CA, FSPTC, and FSAIN) are either all active or all standby.

Because the CA/FS node is likely to be in a “mixed” state, the utility script and the CLI command checks a specific switchover target or all three applications using an optional argument.

Database Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature performs the following tasks to check database synchronization, replication, and shared memory integrity.

1. Audit the DB to check for a mismatch between the EMS and the CA.
2. Query for the following DB replication related alarms:

Type and Number	Description	Severity
Database (3)	There Are Errors in Element Management System Database DefError Queue; Contact Database Administrator (There are Errors in EMS Database DefError Queue; Contact DBA)	Critical
Database (4)	Element Management System Database HeartBeat: Replication Push Job Broken (EMS DB_Heart_Beat: Replication Push Job Broken)	Critical
Database (5)	Element Management System Database HeartBeat Process Died (EMS DBHeartBeat Process Died)	Critical

3. Run the shared memory integrity tool to validate the shared memory integrity of the CA processes.

System Time Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature queries the system for the following alarms to see there is any system time drift.

Type and Number	Description	Severity
Audit (11)	Critical Network Time Protocol Service Failure (Critical NTP Service Failure)	Critical
Audit (12)	Major Network Time Protocol Service Failure (Major NTP Service Failure)	Major
Maintenance (77)	Mate Time Differs Beyond Tolerance	Major

Switchover Impact Alarms Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature queries the database for the following alarms to see if they are being raised by the switchover target side. A complete listing of these outstanding alarms is stored in a log file.

Type and Number	Description	Severity
Call Processing (12)	Feature Server Both Links Down	Critical
Maintenance (50)	Index Table Usage Exceeded Critical Usage Threshold Level (IDX Table Usage Exceeded Critical Usage Threshold Level)	Critical
Maintenance (53)	The Central Processing Unit Usage is Over 90% Busy (The CPU Usage is Over 90% Busy)	Critical
Maintenance (55)	The Five Minute Load Average is Abnormally High	Major
Maintenance (57)	Memory and Swap are Consumed at Critical Levels	Critical
Maintenance (61)	No Heartbeat Messages Received Through the Interface (No HB Messages Received Through the Interface)	Critical
Maintenance (62)	Link Monitor: Interface Lost Communication	Major
Maintenance (63)	Outgoing Heartbeat Period Exceeded Limit (Outgoing HB Period Exceeded Limit)	Major
Maintenance (64)	Average Outgoing Heartbeat Period Exceeds Major Alarm Limit (Average Outgoing HB Period Exceeds Maj Alarm Limit)	Major

Type and Number	Description	Severity
Maintenance (65)	Disk Partition Critically Consumed	Critical
Maintenance (66)	Disk Partition Significantly Consumed	Major
Maintenance (68)	The Free Inter-Process Communication Pool Buffers Below Major Threshold (The Free IPC Pool Buffers Below Major Threshold)	Major
Maintenance (69)	The Free Inter-Process Communication Pool Buffers Below Critical Threshold (The Free IPC Pool Buffers Below Critical Threshold)	Critical
Maintenance (70)	The Free Inter-Process Communication Pool Buffer Count Below Minimum Required (The Free IPC Pool Buffer Count Below Minimum Required)	Critical
Maintenance (82)	Average Outgoing Heartbeat Period Exceeds Critical Limit (Average Outgoing HB Period Exceeds Critical Limit)	Critical
Maintenance (84)	Swap Space Below Major Threshold	Major
Maintenance (85)	Swap Space Below Critical Threshold	Critical
Maintenance (107)	No Heartbeat Messages Received Through Interface From Router (No HB Messages Received Through Interface From Router)	Critical
Signaling (109)	Stream Control Transmission Protocol Association Failure (SCTP Association Failure)	Major
Signaling (113)	Signaling Gateway Failure	Major
Signaling (114)	Signaling Gateway Process is Out-of-Service	Major
Signaling (121)	Message Transfer Part 3 User Adapter Cannot Go Standby (M3UA/SUA Cannot Go Standby)	Major

Inter-Node Communication Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature performs the following communication status checks on all four nodes:

- Checks if internode communication links are established
- Checks if hub is communicating
- Checks if EMS and CA can communicate
- Checks if EMS and feature server can communicate

Process Configuration Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature queries the system for the following alarms to see if there are any feature server configuration errors.

Type and Number	Description	Severity
Configuration (5)	Feature Server Database and Command Line Host Mismatch (Feature Server DB and Command Line Host Mismatch)	Minor

Operating System Issues in /var/adm/messages Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature checks /var/adm/messages for any operating system errors. The feature searches for the following Solaris event types:

- kern.err
- kern.crit
- kern.em

Software Configuration Check

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature checks the following items.

- Verify that the mem.cfg files are identical on primary and secondary EMS nodes
- Verify that the mem.cfg files are identical on primary and secondary CA nodes
- Verify that the patch/version levels are identical on primary and secondary EMS nodes
- Verify that the patch/version levels are identical on primary and secondary CA nodes

Installing

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature utility script is packaged with the Cisco BTS 10200 software upgrade automation scripts. The script resides in the /opt/ems/utils directory after the Cisco BTS 10200 software is installed or upgraded. The script output logs are managed by the check log function, which is run periodically as a cron job.

Command Responses

The new **presw-diag** CLI command internally executes the Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility script. The output of the utility script is displayed as the command response. For security reasons, user input for the password field is masked with asterisks (“*”). The assumption here is that the passwords for of all four Cisco BTS 10200 nodes are identical.

Example:

show presw-diag password=***; [target=appId];

- password field:
  - Root password of the Cisco BTS 10200 nodes
- target field:
  - Is optional parameter, and is used to specify the switchover target.
  - Allowed values are CA, FSPTC, FSAIN.
  - If target field is specified, the mate of the active side of the specified application is checked.
  - If target field is not specified, the mate of the active side of all three applications is checked.

CLI Database

Note We recommend executing the CLI command to utilize the Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility feature.

The **presw-diag** CLI command internally executes the Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility script. The output of the utility script is displayed as the command response. For security reasons, user input of the password field is masked with asterisks ("*"). The assumption here us that the passwords for all four Cisco BTS 10200 nodes are identical. All existing commands used with this feature are documented in the Cisco BTS 10200 Softswitch CLI Database.

Script Arguments

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility script will uses an optional argument for specifying the switchover target. Possible argument values are CA, FSPTC, and FSAIN. The usage of this optional argument is the same as that for the **presw-diag** CLI command target field.

Script Output

The Cisco BTS 10200 prior to manual switchover switch integrity diagnostic utility script displays the resulting summary on the screen and also stores detailed information in a log file.

Log File

The script output log file

- Contains a timestamp so that its name is unique
- Is stored in the /opt/ems/log directory on the node that the script utility is running from
- Is managed by the Cisco BTS 10200 log archiving utility

Result Summary

The script utility uses the following format for the result summary. The summary is displayed on the screen and is stored in the log file.

Legend for the possible values:

- boolean: either "Y" or "N"

Switchover target status okay = boolean

EMS and CA DB in sync = boolean

EMS-A and EMS-B DB in sync = boolean

CA shared memory integrity okay = boolean

System time in sync = boolean

No switchover impact alarms = boolean

Process configuration okay = boolean

Inter-node communication okay = boolean

No issues in OS messages file = boolean

Cisco BTS 10200 software configuration okay = boolean

Log file = [logFileName]

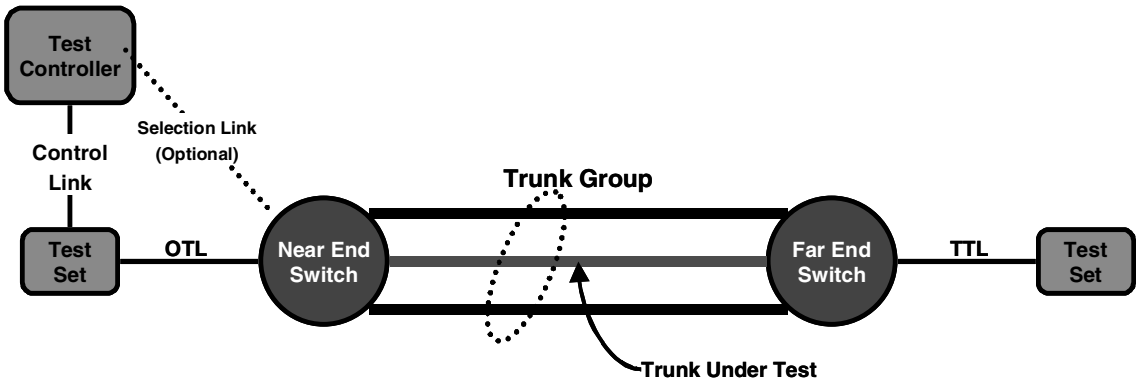
PSTN Trunk Testing

The legacy PSTN trunk network supports connection and performance appraisal testing individual trunks or network routes. This is generally referred to as 100-type tests. The Cisco BTS 10200 provides specific capabilities to support test call origination to selected individual trunks as well as test call termination.

Test Overview

Trunk testing is used to ascertain the transmission quality of the shared trunks used to interconnect switching systems. This is necessary because there is no other practical way to objectively determine each trunk's performance. Figure 15-2 depicts a typical trunk test system.

Figure 15-2 Typical Trunk Test System



The test controller is located on the originating side of the trunk test system. The controller selects a trunk group and a specific trunk within the trunk group to test. It then instructs the near end test equipment, which is connected to the OTL and switch to select the specified trunk and the destination number for the far end test set.

The near end switch then selects the Trunk Under Test (TUT) and, if the TUT is idle, dials the destination number through CAS or nonassociated signaling methods common to normal signaling for the trunk group. If the TUT is busy, an announcement is returned (usually reorder) towards the near end test set and the test call does not proceed.

The far end switch responds to the dialed digits by connecting to the far end test set via the TTL. The far end test set answers the call request. The near end and far end test equipment then conduct the required tests. The results are retrieved by the Test Controller.

The Cisco BTS 10200 supports OTL and TTL capability. User provided test equipment and, optionally, test controllers may be connected to the test lines. Interoperability between different carriers is ensured through proper selection of test equipment and test functions.

For the purposes of PSTN trunk testing, the near end is the Cisco BTS 10200 platform.

Cisco BTS 10200 Originating Test Line

This section discusses the following Cisco BTS 10200 originating test line information:

Function

The OTL originates all test calls. The OTL may be part of an automated trunk test system (for example, CAROT) that will select trunks, make test calls, conduct tests, record measurements and report marginal or inferior trunk performance.

Test Equipment

Test equipment capable of seizing the test line, outpulsing digits (preferably MF format), recognizing supervision, and supporting 1XX tests. While Cisco does not recommend any vendor, SAGE Instruments 930 or 935 series test equipment with the proper options are examples for use.

Test Line

Many gateway products can satisfy the OTL requirements. Preferred capabilities include

- Must be supported by the Cisco BTS 10200.
- T1 access line to connect to the test equipment to minimize transmission impairments caused by codecs and analog filters.
- Preferred signaling arrangement is wink start with MF signaling. (Other signaling arrangements can be supported).

Trunk Access

The Cisco BTS 10200 OTL can logically access up to 9,999 trunk groups, each with up to 9,999 trunks.

Conditions for trunk test access are met when either the requested trunk is in service and idle or the requested trunk is out of service or blocked. Trunk access is denied when the requested trunk is busy. If that happens, route advance is inhibited, and an announcement is returned.

Trunk Access and Test Termination Number Format

Figure 15-3 depicts the dialed digit format for accessing selected trunks and performing tests. These are the digits that the trunk test system or user actually dials. Figure 15-3 shows the format when the OTL is configured for MF signaling.

Figure 15-3 OTL Configured for MF Signaling

Test Type	Test Line	Dialed Digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Comment
Transmission Tests To Standard Test Lines	100	KP	Trunk Group Number				Member Trunk				1	0	0	ST						MW + QT
	101												1							Communications & Test
	102												2							MW
	103												3							Signal/Supervisory
	104												4							2-Way Tests
	105												5							CAROT ROTL/Responder
	N/A																			
	107												7							Data Transmission
	108												8							Digital Loopback
	109												9							Echo

Trunk Under Test Outpulsing

Once the specified trunk is selected, the Cisco BTS 10200 translates the dialed digits into a digit string for outpulsing. Once the trunk under test (TUT) is seized, it will outpulse the destination digits depicted in Figure 15-4. Since the digits may be sent by SS7, MF, or DTMF, only the actual destination digits are depicted.

Figure 15-4 Outpulsed Destination Digits

Test Type	Test Line	Dialed Digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Comment
Transmission Tests	100	9	5	8	1	1	0	0												100 Test Line Group
	101							1												101 Test Line Group
	102							2												102 Test Line Group
	103							3												103 Test Line Group
	104							4												104 Test Line Group
	105							5												105 Test Line Group
	N/A																			N/A
	107							7												107 Test Line Group
	108							8												108 Test Line Group
	109							9												109 Test Line Group

Cisco BTS 10200 Terminating Test Line

This section discusses the following Cisco BTS 10200 terminating test line information:

Function

The TTL terminates all test calls. The TTL may be a responder capable of interacting with an automated trunk test system (for example, CAROT) or it may be a manual test line termination.

Test Equipment

Test equipment must be capable of recognizing an incoming call request from the test line, returning an answer signal, recognizing supervision, and supporting 1XX tests. Although Cisco does not recommend any vendor, SAGE Instruments 930 or 935 series test equipment with the proper options are good choices.

Test Line

Many gateway products can satisfy the OTL requirements. Preferred capabilities include

- Must be supported by the Cisco BTS 10200.
- T1 access line to connect to the test equipment to minimize transmission impairments caused by codecs and analog filters.
- Preferred signaling arrangement is immediate start with no incoming digits. (Other signaling arrangements can be supported).

TTL Dial Plan

The Cisco BTS 10200 test lines are typically assigned 958-11XX numbers as depicted in Figure 15-5. Any line or trunk may dial the appropriate digits to reach a TTL. Other dial plans are also supported and may also work in conjunction with the depicted plan.

Figure 15-5 958-11XX Number Assignment

Test Type	Test Line	Dialed Digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Comment
Transmission Tests	100	9	5	8	1	1	0	0												100 Test Line Group
	101							1												101 Test Line Group
	102							2												102 Test Line Group
	103							3												103 Test Line Group
	104							4												104 Test Line Group
	105							5												105 Test Line Group
	N/A																			N/A
	107							7												107 Test Line Group
	108							8												108 Test Line Group
	109							9												109 Test Line Group

Near End Test Origination Test Line

The BTS 10200 supports calls used to test individual trunks that connect a local gateway with a gateway or PSTN switch at a remote office. The BTS 10200 supports OTL and TTL capability. User-provided test equipment and, optionally, test controllers can be connected to the test lines. Proper selection of test equipment and test functions helps to ensure interoperability between different carriers.

The processes described in this section are applicable to the BTS 10200. The processes might work differently on other switches.

The process for testing a BTS 10200 OTL is as follows:

1. The user verifies that the remote CO has the desired 1xx test line available.
2. The user sets up a test device on a CAS TGW that is connected to the local BTS 10200.
3. The user provisions the CAS-TG-PROFILE table, setting TEST-LINE = YES. (Provisioning commands are described in the Cisco BTS 10200 Softswitch CLI Database.)
4. On the test device at the CAS TGW side, the user enters digits representing the circuit to be tested and the test to be performed:
   - TG, for example 0003
   - Trunk number, for example 0018

     The complete trunk address in this example is 00030018.
   - Test type (10x), for example 104

     The technician dials KP-00030018-104-ST.
5. The BTS 10200 automatically inserts either 9581 or 9591 in front of the test type digits to create a dialing string.

   The complete test string in this example is PREFIX | 00030018 | 9581104 | END.

Note Alternatively, with the BTS 10200, the user can dial the test type with the 9581 or 9591 included: KP-00030018-9581104-ST.

6. The BTS 10200 selects the trunk to be tested based on the user-defined trunk address.
7. The TGW outpulses the digits to the remote switch over the designated trunk.

Far End Originating Test Line

The far end originating test line (OTL) may be located on a different switch product as well as on a different carrier (for example, ILEC, IXC, CLEC). The far end OTL connects to the near end Cisco BTS 10200 softswitch TTL through the TUT. This section discusses the following Cisco BTS 10200 far end originating test line information:

Function

The OTL originates all test calls towards the Cisco BTS 10200 softswitch. The OTL may be part of an automated trunk test system (for example, CAROT) that will select trunks, make test calls, conduct tests, record measurements and report marginal or inferior trunk performance.

Test Equipment

Test equipment capable of seizing the test line, outpulsing digits, recognizing supervision, and supporting 1XX tests. Although Cisco does not recommend any vendor, SAGE Instruments 930 or 935 series test equipment with the proper options are good choices.

Test Line

OTL requirements are specific to the Far End switch product as well as to far end service provider/enterprise test methods and procedures. That subject, however, is outside the scope of this document.

Trunk Access

This is specific to the far end switch product and outside the scope of this document.

Trunk Access and Test Termination Number Format

This is specific to the far end switch product and outside the scope of this document.

Trunk Under Test Outpulsing

The far end switch translates the dialed digits into a digit string for outpulsing. The Cisco BTS 10200 softswitch expects to receive destination digits depicted in Figure 15-6. Since the digits might be sent through SS7, MF, or DTMF, only the actual destination digits are depicted.

Figure 15-6 Received Destination Digits

Test Type	Test Line	Dialed Digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Comment
Transmission Tests	100	9	5	8	1	1	0	0												100 Test Line Group
	101							1												101 Test Line Group
	102							2												102 Test Line Group
	103							3												103 Test Line Group
	104							4												104 Test Line Group
	105							5												105 Test Line Group
	N/A																			N/A
	107							7												107 Test Line Group
	108							8												108 Test Line Group
	109							9												109 Test Line Group

Far End Terminating Test Line

This section discusses the following Cisco BTS 10200 far end terminating test line information:

Function

The TTL terminates all test calls. The TTL may be a responder capable of interacting with an automated trunk test system (for example, CAROT) or it may be a manual test line termination.

Test Equipment

Test equipment capable of recognizing an incoming call request from the test line, returning an answer signal, recognizing supervision, and supporting 1XX tests. Although Cisco does not recommend any vendor, SAGE Instruments 930 or 935 series test equipment with the proper options are good choices.

Test Line

OTL requirements are specific to the far end switch product as well as far end service provider/enterprise test methods and procedures. This is outside the scope of this document.

TTL Dial Plan

Test lines are typically assigned 958-11XX numbers as depicted in Figure 15-7.

Figure 15-7 958-11XX Number Assignments

Test Type	Test Line	Dialed Digits 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Comment
Transmission Tests	100	9	5	8	1	1	0	0												100 Test Line Group
	101							1												101 Test Line Group
	102							2												102 Test Line Group
	103							3												103 Test Line Group
	104							4												104 Test Line Group
	105							5												105 Test Line Group
	N/A																			N/A
	107							7												107 Test Line Group
	108							8												108 Test Line Group
	109							9												109 Test Line Group

1xx Test Lines

This section discusses the following Cisco BTS 10200 1xx test line information:

1xx Test Line Support

When the BTS 10200 is the near-end switch, the following process takes place at the remote switch:

1. The remote switch recognizes the trunk test prefix (9581 or 9591) on the incoming signal, and it uses the test type to route the test to the appropriate test line.
2. The appropriate tests are performed on the test set.
3. Additional test processes may be used, depending on the specific test configuration.

When the BTS 10200 is supporting the TTL capability (test call originated at another switch), the BTS 10200 receives the 958 or 959 call, recognizes the 958 or 959 type, and routes the test to the appropriate test line.

The BTS 10200 enables a TDM-based testing device to perform continuity testing over an MF CAS TDM trunk interface. An MGCP-based trunking gateway must be present in the test path. The TDM test type is the traditional 1xx test type, with an additional enhancement—the ability to route the test call to a specified DN on a given trunk circuit.

100 Test–Balance

The balance test is normally used for two-wire switches to ascertain the performance of the four-wire terminating set "4WTS" or hybrid. Improper options or equipment faults can cause the trunk to sound hollow or have an echo.

This test can also be used to determine the far to near loss of the trunk under test, in some cases, as well as the far to near noise.

When called, the far end test set will either immediately answer with a quiet termination (silence) or provide a milliwatt test tone for a brief period.

101 Test–Communications and Test

This test supports testers to evaluate the TUT by actually talking over it. Normally, the test line is routed to a test position. It also supports manual or specialized testing across the TUT.

102 Test–Milliwatt

The milliwatt test provides a test tone throughout the test. Periodically, the tone may be removed automatically by the far end test set for a brief period of approximately 1 second in every 10 seconds. This helps failed T1 lines to regain frame synchronization and may also be used for other purposes.

This test may be used to determine the far to near loss and/or C-Notched noise of the trunk under test. It may also be used for other far to near test purposes.

103 Test–Signaling and Supervisory

The 103 test provides a connection to a supervisory and signaling test circuit for overall testing of these features on intertoll trunks equipped with ring forward.

104 Test–2-Way Test

Supports far to near and near to far evaluation for the TUT. The operation is very simple with the far end test equipment proceeding through a specific sequence of test steps.

The 104 test supports 2-way transmission testing and 2-way noise checking.

105 Test–ROTL/Responder

This is the preferred test line as it supports many tests for either the near to far or far to near direction. The near end test equipment is normally able to communicate with the far end test equipment to set up and conduct specified tests.

For example, the SAGE 930/935 test sets provide a robust menu of tests that include phase hits, jitter, and nonlinear distortion.

The 105 test line is normally used by CAROT and other automated trunk test systems as the far end test line. In CAROT terms, this is commonly called the responder.

107 Test Line–Data Transmission

The data transmission test line supports 1-way testing of certain voice band data parameters. This includes peak to average ratio signal (PAR), slope, quiet termination, and intermodulation distortion test signals.

It should be noted that newer test equipment, like the SAGE 930/935, provides these and other voice band data tests for *both* directions makes it possible to use one test line to evaluate voice and voice band data performance.

108 Test–Digital Loopback

The 108 test line supports testing by means of a digital loopback. The T108 test line feature determines the performance of trunks connecting digital exchange switches, including voice over packet (VoP) softswitches. BTS 10200 incoming trunks requesting other 1xx-type test lines are routed to shared test lines for the requested tests, regardless of which gateway terminates the trunk or which gateway/IAD terminates the test line. The T108 test line feature requests a test to be performed within the same gateway where the trunk under test (TUT) is terminated, and provides a digital loopback within the gateway. The T108 test line feature supports manual and automated testing.

The T108 test line sequence is as follows:

1. The near-end switch originates the test sequence by placing a test call, identifying the trunk to be selected, and the test line number. A digital test pattern generator is used in the test setup shown in Figure 15-2.
2. The near-end switch uses the trunk identifier to override normal call processing and select only the requested trunk.
3. The far-end switch responds to the destination number and connects to the T108 test line. The T108 test line enables a digital loopback.
4. When the near-end switch receives answer supervision, it conducts digital test sequences to ascertain trunk performance.
5. Once the test sequences are completed, the near-end switch releases the test call and both switches release the trunk connection.
6. The far-end switch can detect if the test connection exceeds a preset time and releases the test connection if the preset time is exceeded.

The T108 test line is also used for trunk redirection (wholesale dial) for Internet services where the carrier modem termination is integrated into the trunk gateway. In this case, the integral digital stored program (DSP) normally supports modem-only transmissions.

109 Test–Echo

The 109 test line supports in-service testing of echo cancellers or echo suppressors.