

PGW 2200 Softswitch: Collect Troubleshooting Information for TAC Cases

Document ID: 27580

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Basic Information

Export the Cisco PGW 2200 Configuration

Further Information Requested by the TAC

Core dumps

Call Traces

- Collect the PGW 2200 MDL Trace
- Collect HSI MDL Traces
- Collect Snooper/Sniffer Traces
- Collect debug Information on the Gateway
- Collect System Data
- Mini Parse Analyzing Tool
- Per Call Trace

Related Information

Introduction

When you open a case with the Cisco Technical Assistance Center (TAC), some preliminary information is required in order to better identify and qualify the issue. Some of this information is always required and other information depends upon the nature of the issue. If you are asked to collect this information by the engineer after you open your case, it results in resolution delay. Also, a good description of the problem is requested, which needs to include the call flow of the way that you ran into the problem at that time. The collection of Message Definition Language (MDL), snooper, sniffer and debug information can be attached to the case notes when you open the case.

The primary goal of this document is to identify the required preliminary information, based on the type of issue. This is so you can provide information to the engineer immediately. The second goal is to provide general guidelines to follow when you collect information for the TAC in order to avoid repetitive testing and recollection of identical data.

This document is intended for Cisco customers that support Data and Voice Signaling Solutions based on Cisco PGW 2200 Softswitch (formerly SC 2200 and VSC 3000, or Cisco Telephony Controller) Media Gateway Controller (MGC) software.

Prerequisites

Requirements

Support personnel need to be familiar with MGC-based solutions and their components. For further information, refer to the links supplied in the Related Information section of this document.

Components Used

The information in this document is based on these software versions:

- Release Notes for the Cisco Media Gateway Controller Software Release

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

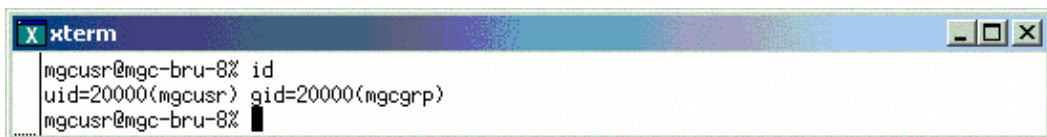
Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Basic Information

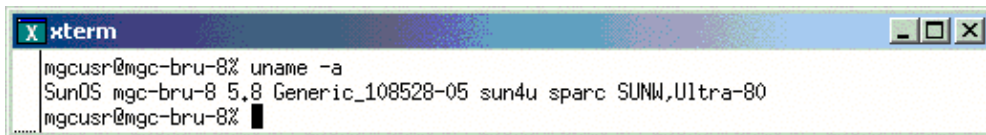
Complete these steps:

1. Before you collect any information, make sure you are logged on as a member of mgcgrp. In order to do this issue the **id** command.



```
xterm
mgcusr@mgc-bru-8% id
uid=20000(mgcusr) gid=20000(mgcgrp)
mgcusr@mgc-bru-8% █
```

2. Issue the **uname** command in order to find out you UNIX version.



```
xterm
mgcusr@mgc-bru-8% uname -a
SunOS mgc-bru-8 5.8 Generic_108528-05 sun4u sparc SUNW,Ultra-80
mgcusr@mgc-bru-8% █
```

3. Issue the **prtconf** command in order to find out the RAM size.



```
xterm
mgcusr@mgc-bru-8% prtconf | grep Memory
Memory size: 4096 Megabytes
mgcusr@mgc-bru-8% █
```

Another way to determine the RAM size is to issue the **uname** command.

```

mgcusr@mgc-bru-8% /usr/platform/`uname -i`/sbin/prtdiag
System Configuration: Sun Microsystems sun4u Netra t 1400/1405 (4 X UltraSPARC-II 440MHz)
System clock frequency: 110 MHz
Memory size: 4096 Megabytes

===== CPUs =====

```

Brd	CPU	Module	Run MHz	Ecache MB	CPU Impl.	CPU Mask
0	0	0	440	4.0	US-II	10.0
0	1	1	440	4.0	US-II	10.0
0	2	2	440	4.0	US-II	10.0
0	3	3	440	4.0	US-II	10.0

```

===== IO Cards =====

```

Brd	Bus Type	Freq MHz	Slot	Name	Model
0	PCI	33	1	network-SUNW,hme	
0	PCI	33	3	scsi-glm/disk (block)	Symbios,53C875
0	PCI	33	3	scsi-glm/disk (block)	Symbios,53C875
0	PCI	33	4	SUNW,hme-pci108e,1001	SUNW,qsi-cheerio

```

No failures found in System
=====
mgcusr@mgc-bru-8%

```

4. Issue the **df** command in order to determine the amount of free disk space available.

```

mgcusr@mgc-bru-8% df -k

```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	1592555	41978	1502801	3%	/
/dev/dsk/c0t0d0s6	1111955	669240	387118	64%	/usr
/proc	0	0	0	0%	/proc
fd	0	0	0	0%	/dev/fd
mnttab	0	0	0	0%	/etc/mnttab
/dev/dsk/c0t0d0s5	494235	99555	345257	23%	/var
swap	3878600	0	3878600	0%	/var/run
/dev/dsk/c0t1d0s7	17413250	9	17239109	1%	/free
swap	3879504	904	3878600	1%	/tmp
/dev/dsk/c0t3d0s0	17408538	6714896	10519557	39%	/data
/dev/dsk/c0t0d0s4	9982616	3049282	6833508	31%	/opt
mgc-bru-7:/data	17413250	15497180	1741938	90%	/data7

```

mgcusr@mgc-bru-8%

```

5. Save the results of the commands described in the earlier steps and redirect the output to the file with either the **> ./filename.txt** command or the **>> ./filename.txt** command for multiple attachments.

In order to collect information about Layer 2 and 3 of the NIC operation, issue the **ifconfig** and **netstat** commands. Some of this information is good for case notes updates.

```

mgcusr@mgc-bru-8% ifconfig -a > ./netinfo.txt
mgcusr@mgc-bru-8% netstat -i >> ./netinfo.txt
mgcusr@mgc-bru-8% netstat -rn >> ./netinfo.txt
mgcusr@mgc-bru-8% more ./netinfo.txt
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ffffffff
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.48.84.87 netmask ffffffff broadcast 10.48.84.255
hme1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 10.48.85.87 netmask ffffffff broadcast 10.48.85.255
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 240748 0 240748 0 0 0
hme0 1500 mgc-bru-8 mgc-bru-8 19424559 0 14345041 0 0 0
hme1 1500 mgc-bru-8b mgc-bru-8b 1704841 0 117754 0 1 0

Routing Table: IPv4
Destination Gateway Flags Ref Use Interface
-----
10.48.84.0 10.48.84.87 U 1 30729 hme0
10.48.85.0 10.48.85.87 U 1 1512 hme1
224.0.0.0 10.48.84.87 U 1 0 hme0
default 10.48.84.1 UG 1 4057
127.0.0.1 127.0.0.1 UH 24 240306 lo0
mgcusr@mgc-bru-8% █

```

6. The netinfo.txt file that is created needs to be sent to the TAC engineer. The patch level is checked with the use of the **pkginfo** command with the output redirected to a file.

```

mgcusr@mgc-bru-8% pwd
/opt/CiscoMGC/local
mgcusr@mgc-bru-8% pkginfo | grep CSC0 > ./patches-installed.txt
mgcusr@mgc-bru-8% more patches-installed.txt
application CSC000004 Cisco Media Gateway Controller Software core system protocol
application CSC001004 Cisco Media Gateway Controller Software Miscellaneous Protoc
application CSC002004 Cisco Media Gateway Controller Software Miscellaneous Protoc
application CSC010004 Cisco Media Gateway Controller Software SS7 family protocols
application CSC020004 Cisco Media Gateway Controller Software PRI family protocols
application CSC021004 Cisco Media Gateway Controller Software PRI family protocols
application CSC030004 Cisco Media Gateway Controller Software Q761 Version 1 famil
application CSC031004 Cisco Media Gateway Controller Software Q761 Version 1 famil
application CSC032004 Cisco Media Gateway Controller Software Q761 Version 1 famil
application CSC033004 Cisco Media Gateway Controller Software Q761 Version 1 Famil
application CSC040004 Cisco Media Gateway Controller Software Q761 Version 2 Proto
application CSC041004 Cisco Media Gateway Controller Software Q761 Version2 Protoc
application CSC050004 Cisco Media Gateway Controller Software Q761 Version 3 Proto
application CSC060004 Cisco Media Gateway Controller Software Q721 Protocol Family
application CSC070004 Cisco Media Gateway Controller Software Q767 Protocol Family
application CSC071004 Cisco Media Gateway Controller Software Q767 Protocol Family
application CSC080004 Cisco Media Gateway Controller Software Q931 Protocol Family
application CSC0ga000 Cisco Media Gateway Controller Software Licensing Components
application CSC0ga001 Cisco Media Gateway Controller Software Application
application CSC0ga002 Cisco Media Gateway Controller Software Database Components
application CSC0ga003 Cisco Media Gateway Controller Software SNMP Components
application CSC0ga004 Cisco Media Gateway Controller Software Toolkit Applications
application CSC0gc001 Cisco Media Gateway Controller Software Configuration & Data
application CSC0gs004 Cisco Media Gateway Controller Software Patch Package
application CSC0gt001 Cisco Media Gateway Controller Software Analyzer/Simulator
utilities CSC0gu000 Cisco Media Gateway Controller Software Common Utilities & L
utilities CSC0h015 Media Gateway Controller Solaris 8 Patch Cluster
application CSC0vsp22 Cisco Voice Services Provisioning Tool
mgcusr@mgc-bru-8% █

```

7. The patches-installed.txt file that is created needs to be sent to the TAC engineer. Issue the **mml** command to see the PGW 2200 software version that currently runs.

```
xterm
mgc-bru-8 mml> rtrv-ne
MGC-01 - Media Gateway Controller 2002-09-26 16:36:21.415 WET
M RTRV
  "Type:MGC"
  "Hardware platform:sun4u sparc SUNW,Ultra-80"
  "Vendor:"Cisco Systems, Inc.""
  "Location:MGC-01 - Media Gateway Controller"
  "Version:"9.2(2)""
  "Platform State:ACTIVE"
  ;
mgc-bru-8 mml>
```

8. Issue the **rtrv-ne-health** command in order to see the overall health of the network element. This applies to Release 9.

```
xterm
mgc-bru-8 mml> rtrv-ne-health
MGC-01 - Media Gateway Controller 2002-10-17 12:00:47.288 WET
M RTRV
  "Platform State:ACTIVE"
  "Machine Congestion Level = MCL 0 (No Congestion)"
  "Current in progress calls = 0, call attempts = 0 cps"
  "CPU 0 Utilization = 0 % CPU 1 Utilization = 0 %"
  "CPU 2 Utilization = 0 % CPU 3 Utilization = 0 %"
  "Memory (KB): 3917424 Free virtual, 8390328 Total virtual, 4194304 Total real"
  "Filesystem          kbytes  used  avail capacity Mounted on"
  "/dev/dsk/c0t0d0s0  1592555  41981 1502798    3%  /"
  "/dev/dsk/c0t0d0s4  9982616 3314786 6568004   34%  /opt"
  ;
mgc-bru-8 mml>
```

Export the Cisco PGW 2200 Configuration

In most cases, the current Cisco PGW 2200 Softswitch configuration that the TAC requests is used to analyze it for configuration errors. This configuration is not the output of the **prov-rtrv:all** command, as it does not show the provisioning commands that are used. Instead, TAC needs either the configuration script for the problem recreation or the current *.dat files from the system.

1. In order to export and save the current configuration, issue the **prov-exp:all** command from MML.

The **config-exported.tar** file that is created needs to be sent to the TAC engineer.

```
xterm
mgcusr@mgc-bru-8% mml -s8

Copyright © 1998-2001, Cisco Systems, Inc.

mgc-bru-8 mml> prov-exp;all;dirname="config-exported"
  MGC-01 - Media Gateway Controller 2002-09-27 15:24:00.221 WET
M  RTRV
  "ALL"
  ;
mgc-bru-8 mml> quit
mgcusr@mgc-bru-8% cd /opt/CiscoMGC/etc/cust_specific/config-exported/
/opt/CiscoMGC/etc/cust_specific/config-exported
mgcusr@mgc-bru-8% ls -al
total 110
drwxrwxr-x  2 mgcusr  mgcgrp   512 Sep 27 15:24 .
drwxrwxr-x 57 mgcusr  mgcgrp  1024 Sep 27 15:23 ..
-rw-rw-r--  1 mgcusr  mgcgrp  3966 Sep 27 15:24 bru8.mml
-rw-rw-r--  1 mgcusr  mgcgrp  6939 Sep 27 15:23 config.mml
-rw-rw-r--  1 mgcusr  mgcgrp  3646 Sep 27 15:23 export_trkgrp.dat
-rw-rw-r--  1 mgcusr  mgcgrp 15789 Sep 27 15:23 export_trunk.dat
-rw-rw-r--  1 mgcusr  mgcgrp   87 Sep 27 15:24 numan.mml
-rw-rw-r--  1 mgcusr  mgcgrp  128 Sep 27 15:24 portednum.dat
-rw-rw-r--  1 mgcusr  mgcgrp 17321 Sep 27 15:23 properties.dat
-rw-rw-r--  1 mgcusr  mgcgrp  2090 Sep 27 15:24 routing.mml
mgcusr@mgc-bru-8% tar cfv config-exported.tar ./*.dat
a ./bru8.mml 4K
a ./config.mml 7K
a ./export_trkgrp.dat 4K
a ./export_trunk.dat 16K
a ./numan.mml 1K
a ./portednum.dat 1K
a ./properties.dat 17K
a ./routing.mml 3K
mgcusr@mgc-bru-8% █
```

2. Use the tar archiving utility in UNIX in order to save current *.dat files.

The config.tar file that is created needs to be sent to the TAC engineer who works on the case.

```
xterm
mgcusr@mgc-bru-8% pwd
/opt/CiscoMGC/etc
mgcusr@mgc-bru-8% tar cvf ./config.tar ./*.dat
a ./accRespCat.dat 1K
a ./alarmCats.dat 41K
a ./alarmTable.dat 3K
a ./auxSigPath.dat 0K
a ./bearChan.dat 0K
a ./bearChanSwitched.dat 23K
a ./buckets.dat 3K
a ./cable.dat 0K
a ./charge.dat 0K
a ./chargeholiday.dat 0K
a ./components.dat 11K
a ./compTypes.dat 4K
a ./condRoute.dat 0K
a ./crossConnect.dat 0K
a ./dependencies.dat 1K
a ./dialplan.dat 1K
a ./digitAnalysis.dat 1K
a ./dmprSink.dat 1K
a ./dns.dat 1K
a ./dpc.dat 1K
a ./extNodes.dat 1K
a ./extNodeTypes.dat 1K
a ./extProcess.dat 1K
a ./files.dat 1K
a ./gtdParam.dat 0K
a ./linkSetProtocol.dat 1K
a ./mclCallReject.dat 1K
a ./mclThreshold.dat 1K
a ./mdlProcess.dat 1K
a ./measCats.dat 56K
a ./measProfs.dat 2K
a ./mmlCommands.dat 3K
a ./physLineIf.dat 1K
a ./processes.dat 3K
a ./procGroups.dat 1K
a ./profileComps.dat 0K
a ./profiles.dat 0K
a ./profileTypes.dat 1K
a ./properties.dat 17K
a ./propSet.dat 54K
a ./routeAnalysis.dat 4K
a ./routes.dat 1K
a ./services.dat 5K
a ./sigChanDev.dat 1K
a ./sigChanDevIp.dat 1K
a ./sigPath.dat 1K
a ./snmpmgr.dat 0K
a ./stp.dat 0K
a ./tables.dat 2K
a ./tariff.dat 0K
a ./testLine.dat 1K
a ./thresholds.dat 2K
a ./trigger.dat 47K
a ./trunkGroup.dat 1K
a ./variant.dat 4K
a ./version.dat 1K
a ./XECfgParm.dat 16K
mgcusr@mgc-bru-8% □
```

Further Information Requested by the TAC

If the problem is not solved at this stage, more information is required. Collect the information described in this section on log messages and alarms.

1. Check for OS Solaris messages. File messages and messages.x needs to be sent to the TAC engineer.

```
xterm
mgcusr@mgc-bru-8% pwd
/var/adm
mgcusr@mgc-bru-8% ls -al
total 182464
drwxrwxr-x 7 root sys 512 Sep 22 03:10 .
drwxr-xr-x 30 root sys 512 Feb 27 2002 ..
-rw-r--r-- 1 uucp bin 0 Feb 27 2002 aculog
drwxr-xr-x 2 adm adm 512 Feb 27 2002 exacct
-r--r--r-- 1 root other 560028 Sep 27 15:12 lastlog
drwxr-xr-x 2 adm adm 512 Feb 27 2002 log
-rw-r--r-- 1 root other 9175469 Sep 27 15:30 messages
-rw-r--r-- 1 root other 11627227 Sep 22 03:09 messages.0
-rw-r--r-- 1 root other 11785962 Sep 15 03:09 messages.1
-rw-r--r-- 1 root other 12144770 Sep 8 03:10 messages.2
-rw-r--r-- 1 root other 11819183 Sep 1 03:07 messages.3
-rw-r--r-- 1 root other 994 Feb 27 2002 MGC_install.log
drwxr-xr-x 2 adm adm 512 Feb 27 2002 passwd
drwxr-xr-x 2 root sys 512 Feb 27 2002 sm.bin
-rw-rw-rw- 1 root bin 0 Feb 27 2002 spellhist
drwxr-xr-x 2 root sys 512 Feb 27 2002 streams
-rw-r--r-- 1 root root 12258 Sep 27 15:18 sulog
-rw-r--r-- 1 root bin 5952 Sep 27 15:12 utmpx
-rw-r--r-- 1 root root 5736 Sep 6 15:02 vold.log
-rw-r--r-- 1 adm 36709704 Sep 27 15:12 wtmpx
mgcusr@mgc-bru-8% █
```

2. Issue the **tail** command in order to check for Cisco PGW 2200 Softswitch application messages.

All platform.log files related to the problem need to be sent to the TAC engineer.

```
xterm
mgcusr@mgc-bru-8% pwd
/opt/CiscoMGC/var/log
mgcusr@mgc-bru-8% tail -f ./platform.log
Mon Sep 30 14:56:14:950 2002 WET | ss7-i-1 (PID 12052) <Error>
sm_writeSession; ssetId 1 is in OOS, PDU dropped

Mon Sep 30 14:56:14:950 2002 WET | ss7-i-1 (PID 12052) <Error>
PROT_ERR_MTP3: MTP3 Error Message from compId c7iplnk1-ibanez1[001d0003]; CP Err
or sending Connect Req

Mon Sep 30 14:56:20:661 2002 WET | ss7-i-1 (PID 12052) <Error>
sm_writeSession; ssetId 1 is in OOS, PDU dropped

Mon Sep 30 14:56:20:661 2002 WET | ss7-i-1 (PID 12052) <Error>
PROT_ERR_MTP3: MTP3 Error Message from compId c7iplnk1-ibanez1[001d0003]; CP Err
or sending Mgmt Statistics Req^C
mgcusr@mgc-bru-8% █
```

3. Issue the **rtvr-alsms** command in order to check for Cisco PGW 2200 Softswitch alarms.

```

X xterm
mgc-bru-8 mml> rtrv-alm::cont
MGC-01 - Media Gateway Controller 2002-09-26 16:43:07.746 WET
M RTRV
"ls-td20: 2002-09-23 08:43:30.401 WET,ALM=\\"FAIL\\",SEV=MJ"
"ss7-td20: 2002-09-23 08:43:30.410 WET,ALM=\\"SS7 SIG SRVC UNAVAIL\\",SEV=MJ"
"c7iplnk1-ls-stp1: 2002-09-23 08:50:00.124 WET,ALM=\\"CHAN BAD TOT 15\\",SEV=MN"
"c7iplnk1-ls-stp1: 2002-09-23 08:55:00.134 WET,ALM=\\"CHAN BAD TOT 60\\",SEV=MN"
"c7iplnk1-ls-stp1: 2002-09-23 10:40:00.344 WET,ALM=\\"CHAN BAD TOT 24\\",SEV=MN"
"c7iplnk1-ls-stp2: 2002-09-23 08:45:00.116 WET,ALM=\\"CHAN BAD TOT 15\\",SEV=MN"
"c7iplnk1-ls-stp2: 2002-09-23 08:55:00.134 WET,ALM=\\"CHAN BAD TOT 60\\",SEV=MN"
"c7iplnk1-ls-stp2: 2002-09-23 09:40:00.224 WET,ALM=\\"CHAN BAD TOT 24\\",SEV=MN"
"c7iplnk1-ibanez1: 2002-09-23 08:43:30.400 WET,ALM=\\"SC FAIL\\",SEV=MJ"
"ss-ibanez1-1: 2002-09-24 11:40:39.760 WET,ALM=\\"IP CONNECTION FAILED\\",SEV=MJ"
;

/* Listening for alarm events... (Ctrl-C to stop) */
^C
/* Ctrl-C pressed */
mgc-bru-8 mml>

```

If you want to collect alarms from log files from the CLI with readable dates and times, create a reprocessing script in Perl (cut and paste this script):

```

alias rdalm 'perl -F, -anwe \'\'print
unpack("x4 A15", localtime($F[1])), ".$F[2]: @F[0,3..7]"\'\'\'

```

CD into the Alarms directory:

```

cd /opt/CiscoMGC/var/spool

```

Execute this command:

```

rdalm alm_yyyymmdd*

```

where `yyymmdd` is the relevant date.

4. Issue the `rtrv-dest:all` command in order to check the basic SS7 network status.

```

X xterm
mgc-bru-8 mml> rtrv-dest:all
MGC-01 - Media Gateway Controller 2002-10-17 11:59:34.070 WET
M RTRV
"ss7-sw1:PKG=SS7-ITU,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"ss7-na1:PKG=SS7-ITU,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"ss7-td20:PKG=SS7-ITU,ASSOC=SWITCHED,PST=OOS,SST=FLD"
"ss7-tdtv2:PKG=SS7-ITU,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"ss7-sw2:PKG=SS7-ITU,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"pri-90:PKG=ISDNPRI,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"pri-91:PKG=ISDNPRI,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"pri-92:PKG=ISDNPRI,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"pri-94:PKG=ISDNPRI,ASSOC=SWITCHED,PST=IS,SST=RSTO"
"pri-93:PKG=ISDNPRI,ASSOC=SWITCHED,PST=IS,SST=RSTO"
;
mgc-bru-8 mml>

```

Core dumps

In case a core dump is found under the directory `/opt/CiscoMGC/var`, run the two UNIX commands `pstack` and `pmmap` under Solaris 2.8 and attach the output to the case notes. Also upload the core file to the case notes.

Note: At Cisco the core file is analyzed via other tools:

```
Example :  
Under /opt/CiscoMGC/var  
<logging as root>  
#pstack <core_file>  
#pmap <core_file>
```

Call Traces

In order to avoid asking "call traces" which give a more granular tracing option on the PGW 2200 for the 'call engine process state' of that call, the MDL tracer gives a C++ object instantiated by the engine that contains a collection of TraceFile objects for that call. This MDL trace can give some details where the problem is related on the PGW 2200 and is welcome during the handling of the case. The best scenario is to upload as much information as possible during the time you encounter the problem. Based on the scenario or solution you run into, you can collect the information detailed in these sections:

Collect the PGW 2200 MDL Trace

Use this procedure in order to collect an MDL trace via the MML command **STA-SC-TRC** (Start Trace).

Based on which Cisco PGW 2200 Softswitch version you run, detailed information can be found for:

- MML Commands for PGW 2200 Release 7.x
- MML Commands for PGW 2200 Release 9.x

1. Identify the Originating SS7 SigPath Number or the Originating TrunkGroup Number on which calls are placed.
2. Rotate the log: run script under **/opt/CiscoMGC/bin/log_rotate.sh**.
3. Start the MDL trace:

```
mml>sta-sc-trc:<ss7sigPath name | orig trunkgroup number>:CONFIRM
```

4. Perform a test (make a call).
5. Stop the MDL trace:

```
mml>stp-sc-trc:all
```

6. Identify the Call Id (C:) of the bad call.

If this test call is made in a test environment, only one CALL_ID displays.

Note: These files can contain tracings from many calls that are all mixed up together if the capture is taken on a production PGW. Each tracing record in the file has a specific record type and records information of a type that relates to that record. Each record has a Call ID that relates it to a specific call.

7. Convert the MDL trace into a readable format:

- a. Go to the **/opt/CiscoMGC/var/trace** directory.
- b. Run this command:

```
get_trc.sh <trace file name>
```

For example:

```
/opt/CiscoMGC/var/trace  
mgcusr@mgc-bru-20%get_trc.sh __ss7path_20040116103221.btr
```

```

get_trc.sh ca/sim/sp Trace File Utility Mistral Version 1.2
The ANALYSIS mdo file is:  GENERIC_ANALYSIS.mdo
Retrieving _ss7path_20040116103221.btr trace file Call ID's, please wait.
Enter one of the following commands:
S = Simprint in less
F = Simprint with printing of sent and received Fields in less
D = Display trc trace in less
G = Display trc trace in less (Generated)
C = Convert to trc trace file
A = Display CA file in less
N = Move to Next call ID
P = Move to Previous call ID
L = List call ID's in current file
X = Set SP flags
H = Print Help
Q = Quit get_trc.sh
Or just enter the ID of the call you want if you know it
Use (N)ext and (P)revious to move between the call ID's
_ss7path_20040116103221.btr contains 1 call(s)
==> Working on call 1 ID 23 H = Help [S/F/D/G/C/A/N/P/L/H/Q/id]?

```

8. Type **Call Id** at the prompt in order to jump to the MDL trace of the bad call.
9. Choose **option C** in order to convert the trace file.

Note: The .btr files are binary trace files that are produced by the PGW tracer function. The main part of the file name is given in the VSC MML command **sta-sc-trc**. The PGW always adds a .btr extension to these files. By using the C option, the file is converted into a text format and the extension has .trc files that are text trace files. They contain detailed line by line trace information from the MDO code that is run in the simulation replay that produces the file. Therefore, they contain MDL traces.

10. The trace file is in **/opt/CiscoMGC/var/trace**.
11. Collect the platform.log information under **/opt/CiscoMGC/var/log**. In some cases the TAC engineer can ask for other platform.log information related to the problem that is reported while the TAC case is handled.

Collect HSI MDL Traces

Refer to HSI Data Collection for Technical Support Service Requests for information and procedures on how to collect HSI MDL traces.

Collect Snooper/Sniffer Traces

Use this procedure to collect snooper/sniffer traces if you have installed Packet Telephony Center – Monitoring and Troubleshooting (PTCMT), or you run an old version of snooper (which is helpful in order to have a good understanding of the call flow).

1. Run **snoop** on all Solaris platforms.
 - a. In order to collect UNIX snoop information, log in as a superuser and run the command **snoop -x42 -o snoop.log <IP address>**.
 - b. Press **Ctrl +C** to exit snoop.
 - c. Upload the **snoop.log** file to the case-notes.

Note: Explain in the case notes that this file has been captured via the UNIX **snoop** command.

2. Run the Cisco **snooper** application.

- a. In order to collect Cisco snoop information, log in as a superuser and run the command **./snooper int INTERFACE PARMS LIST** or you can run **./snooper** which provides a full description.
- b. For the Nailed solution, run **./snooper int hme'x' isdn ss7 rlm > snooper_int1**, where *x* is the interface number. You can also find this when you issue the **ifconfig -a** command. Also, upload the **snooper_int1** file to the case notes.
- c. For the Switched solution, enter **./snooper int hme'x' mgcp ss7 eisup >snooper_int1**, where *x* is the interface number. You can also find this when you issue the **ifconfig -a** command. Also, upload the **snooper_int1** file to the case notes.
- d. For cases where output from two interfaces needs to be captured, use this approach:

```
% ( snooper int hme0 rudp & ; sleep 1 ; snooper int hme1 rudp & ) >> test
% ps -ef | grep snooper | grep -v grep
root 10748 10737 1 20:52:54 pts/15 0:00 snooper int hme1 rudp
root 10736 1 1 20:52:53 pts/15 0:00 snooper int hme0 rudp
% tail -f test
```

3. Run **PTCMT**. For more information, refer to Cisco Packet Telephony Center – Monitoring and Troubleshooting.

In order to collect PTCMT information, log in as a superuser and run the command **./ptcmt int INTERFACE PARMS LIST** or you can run **./snooper** which provides a full description.

- ◆ For the Nailed solution, enter **./ptcmt int hme'x' isdn ss7 rlm > snooper_int1**, where *x* is the interface number. You can also use the **ifconfig -a** command. Also, upload the **snooper_int1** file to the case notes.
- ◆ For the Switched solution, enter **./ptcmt int hme'x' mgcp ss7 eisup >snooper_int1**, where *x* is the interface number. You can also use the **ifconfig -a** command. Also, upload the **snooper_int1** file to the case notes.

Collect debug Information on the Gateway

Based on if you use a Nailed solution [Ni2+] or a Switched solution [MGCP], debug information between the PGW 2200 and the gateway can provide detailed information of the reported problem.

- For the Nailed solution, enter the **debug isdn q931** command and upload the details to the TAC case.
- For the Switched solution, enter the **debug mgcp packet** command and upload the details to the TAC case.

Note: Be aware that you do not run this **debug** command while the CPU load is above 60 percent. You can check this with the **show proc cpu** command. Also, based on the problem that is reported, other **debug** commands can be requested by the TAC engineer.



Caution: Collection of call traces can impact system performance and calls can be dropped. Call traces on a live system need to be done only at the request of the TAC engineer in case you are not sure about the collection of the log information.

Collect System Data

Cisco PGW 2200 Softswitch software includes a data collection script. When you run this script, a data snapshot of your system is saved to a log file. You should run this script shortly after you discover a problem and prior to taking any corrective action. Refer to Collecting System Data for Cisco TAC for more information.

Here is the collect data script:

```
mssol-pgw-6% collectdata
the location of the log file is /opt/CiscoMGC/var/log/200806111552.mssol-pgw-6.log
mssol-pgw-6% ls -al /opt/CiscoMGC/var/log/200806111552.mssol-pgw-6.log
-rw-rw-r-- 1 mgcusr mgcgrp 266375 Jun 11 15:52 /opt/CiscoMGC/var/log/200806111552.m
mssol-pgw-6% more /opt/CiscoMGC/var/log/200806111552.mssol-pgw-6.log
```

Mini Parse Analyzing Tool

The **mini_parse.pl** is a tracing tool that can provide a detailed analysis of call flows. This tool is located in the `/opt/CiscoMGC/bin` folder. Mini-parse provides a simple flow diagram of events that are contained in an MDL trace file (`.trc`). The usage is shown here:

```
mini_parse.pl [-d] [-b] [-i] [-m] [-s] <tracefile>
```

where

- `-d`: adds additional message decode
- `b`: adds additional B number analysis info
- `m`: prints messages only (no internal signals)
- `i`: adds additional IN info
- `s`: adds state transitions

The output can be redirected to a file.

Per Call Trace

You can perform advanced call traces with Cisco PGW 2200 Softswitch Release 9.7(3) Patch 8 and later. The advanced call trace is based on the existing call trace function and adds the calling party number, the called party number, the machine congestion level (MCL) setting, the cause value, and the call duration as call trace criteria. This enhancement makes the call trace more accurate and reduces system performance impacts on the Cisco PGW 2200 Softswitch when the Cisco PGW 2200 Softswitch performs call traces.

For information on how to start the call trace, refer to the Starting A Call Trace (on Release 9.7(3) Patch 8) section of the *Cisco PGW 2200 Softswitch Release 9 Operations, Maintenance, and Troubleshooting Guide*.

Related Information

- [Necessary Skills for Support Personnel](#)
- [Cisco PGW 2200 Softswitch Tech Notes](#)
- [PGW 2200 Release Notes](#)
- [Cisco Signaling Controllers Technical Documentation](#)
- [Voice Technology Support](#)
- [Voice and IP Communications Product Support](#)
- [Troubleshooting Cisco IP Telephony](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 20, 2006

Document ID: 27580
