



CHAPTER 16

Disaster Recovery Procedures

Revised: August 10, 2011, OL-25016-01

Introduction

This chapter tells you how to recover your database in a disaster situation, how to recover your database from another database, and how to recover data from the Call Agent shared memory.

We recommend backing up all data on the Element Management System (EMS), Call Agent (CA) and Feature Server (FS) platforms on a daily basis, and saving the backed up data to a remote server. Data back up files are needed in the unlikely event that data on both the primary and secondary sides of any platform become corrupted. In that case, data must be restored from a back up file.

Restarting a Cisco BTS 10200 Softswitch Process

When a Cisco BTS 10200 Softswitch process exits due to an internal error (such as SIGSEGV on UNIX) or is terminated by the platform, the system automatically restarts the process that shut down.

Restarting the process is a preferred alternative to switching over to the mate, because the restart preserves stable calls and also attempts to preserve transient calls. When a process is restarted, the process audits information such as resource states and attempts to repair inconsistencies. If a process experiences a high failure rate (even after repeated restarts), the system will switch over to the mate.

Disaster Recovery From Flash Archive

This section describes the steps needed to restore the flash archive on the Cisco BTS 10200 system. The flash archive back up is performed before any software upgrade or for maintenance routine purpose. This procedure is used only when both mirrored disks are corrupted or cannot be booted.

Flash archive is a Sun Solaris tool that allows you to take an image of a host and store it on a network file server (NFS) that can be used later for disaster recovery.

For the Cisco BTS 10200, it is best to take a system flash archive whenever the Solaris Operation System is being modified.

Before You Begin

Before restoring your system, you must have the following:

- Bootable Sun Solaris 10 Operating System CD #1



Note Sun Solaris 10 can be download at <http://www.sun.com>.

- Console access
- Restored Host name
- Internet Protocol (IP) address and netmask of restored system
- Location of an archive
- Enabling negotiation on the 2900 switch for the primary interface of the system

Example:

```
c2924.118-A# config t
c2924.118-A(config-t)#int fastEthernet 0/1
c2924.118-A(config-if)#no speed 100
c2924.118-A(config-if)#no full duplex
```

Flash Archive Restore



Note We recommend running this procedure during a maintenance window or when traffic is low.

- Step 1** Connect to the console of the restored unit.
- Step 2** Load the bootable Solaris-10 CD into the compact disk–read only media (CD-ROM) drive.
- Step 3** At the *ok*> prompt, type: **boot cdrom**
- Step 4** Enter **0** for English.
- Step 5** Enter **14** for Other.
- Step 6** Enter **vt100** for terminal type.
- Step 7** Press **Esc-2** to continue.
- Step 8** Press **Esc-2** again to continue.

- Step 9** Press **Esc-2** to continue; use default setting (Mark **X** on *Yes* for Networked).
- Step 10** Choose primary interface and then **Esc-2** to continue.
- Step 11** Press **Esc-2** to continue, use default setting (Mark **X** on *No* for Use Dynamic Host Configuration Protocol (DHCP)).
- Step 12** Enter <hostname> and press **Esc-2** to continue.
- Step 13** Enter <IP address> and press **Esc-2** to continue.
- Step 14** Press **Esc-2** to continue; use default setting (Mark **X** on *Yes* for System part of a subnet).
- Step 15** Enter <Netmask> and press **Esc-2** to continue.
- Step 16** Press **Esc-2** to continue; use default setting (Mark **X** on *No* for Enable IPv6).
- Step 17** Confirm the network information and press **Esc-2** to continue.
- Step 18** Press **Esc-2** to continue; use default setting (Mark **X** on *No* for Configure Kerberos Security).
- Step 19** Press **Esc-2** to continue.
- Step 20** Mark **X** on *None* for Name service. Press **Esc-2** to continue.
- Step 21** Confirm the information and press **Esc-2** to continue.
- Step 22** Choose Continents and Oceans then **Esc-2**.
- Step 23** Choose Countries and Regions then Press **Esc-2**.
- Step 24** Mark **X** on Timezone. Press **Esc-2** to continue.
- Step 25** Set date and time. Press **Esc-2** to continue.
- Step 26** Confirm the information and press **Esc-2** to continue.
- Step 27** Choose **F4** for Flash installation and then press **Esc-4**.
- Step 28** Choose **Manual reboot** and then Press **Esc-2**.
- Step 29** Mark **x** on NFS for NFS Flash Archive Retrieval Method then Press **Esc-2**.
- Step 30** Provide the location of the archive, as shown in the following example and Press **Esc-2**:
- ```
10.89.224.1:/archive/prical8.archive
```
- Step 31** Mark **x** on primary Disk and then Press **Esc-2**.
- Step 32** Press **Esc-2** to continue without preserving data.
- Step 33** Press **Esc-4** for Customize disk layout.
- Step 34** Partition the disk as follow and Press **Esc-2**.
- ```
fileys rootdisk.s0 2000 /
fileys rootdisk.s1 5000 /var
fileys rootdisk.s3 4000 swap
fileys rootdisk.s4 24
fileys rootdisk.s5 free /optfileys rootdisk.s6 2000
```
- Press **Esc-2** to confirm Disk Layout.
- Step 35** Press **Esc-2** to continue.
- Step 36** Press **Esc-2** to continue without remote mounts.
- Step 37** Press **Esc-2** to continue with installation.

**Note**

The restoration takes about 15–30 minutes.

- Step 38** Press “!” (exclamation sign) to exit if prompted.
- Step 39** Verify `/a/etc/vfstab` and `/a/etc/system` files contain no disk mirroring information.

Example of `/a/etc/vfstab`:

```
#####
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
#/dev/dsk/c1d0s2 /dev/rdisk/c1d0s2 /usr ufs 1 yes -
fd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c1t0d0s3 - - swap - no -
/dev/dsk/c1t0d0s0 /dev/rdisk/c1t0d0s0 / ufs 1 yes -
/dev/dsk/c1t0d0s1 /dev/rdisk/c1t0d0s1 /var ufs 1 yes -
/dev/dsk/c1t0d0s5 /dev/rdisk/c1t0d0s5 /opt ufs 2 yes -
swap - /tmp tmpfs - yes -
#####
```

`/a/etc/system` file should not have the following similar lines:

```
#####
* Begin MDD root info (do not edit)
rootdev:/pseudo/md@0:0,2,blk
* End MDD root info (do not edit)
#####
```

- Step 40** Enter the following command:
- ```
cp /a/bin/date /a/bin/.date
mv /a/bin/date.archive /a/bin/date
mv /a/etc/rc3.d/S99platform /a/etc/rc3.d/saved.S99platform
```

- Step 41** Restore 2900 switch back to force 100MB full-duplex.
- Step 42** Power cycle the system.
- 

## Setting Up Interfaces

Use the following procedures to set up the interfaces.

---

- Step 1** Log in as root.
- Step 2** Sftp the following files from the mate:
- ```
cd /tmp
sftp <mate ip address>
get /etc/resolv.conf
get /etc/hosts host
get /etc/netmasks
get /etc/nsswitch.conf
get /etc/default/init
bye
```
- Step 3** Copy the `nsswitch.conf` file to `/etc/` directory:
- ```
cp -p nsswitch.conf /etc/
```

- Step 4** Copy the nsswitch.conf file to /etc/ directory:  
`cp -p nsswitch.conf /etc/`
- Step 5** Copy the resolv.conf file to /etc/ directory:  
`cp -p resolv.conf /etc/`
- Step 6** Copy init file to /etc/default/ directory:  
`cp -p init /etc/default/`
- Step 7** Set up interfaces:  
`cd /opt/setup`  
`setlogic_EMS.sh` (Run this script to set up interfaces on EMS box)  
`setlogic_CA.sh` (Run this script to set up interfaces on CA box)
- Verify all interfaces are setting up properly.
- Step 8** Set up root password by enter the following command:  
`passwd root`
- Step 9** Reboot the system:  
`shutdown -y -g0 -i6`

## Restoring the Cisco BTS 10200 Application

To restore the software application, perform the following steps:

- Step 1** Log in as root.
- Step 2** Run checkCFG script to make sure no errors are encounter.
- Step 3** Restore platforms shared-memory.
- For CA/FS:**
- ```
<hostname># mount <NFS server ip>:/<shared directory> /mnt
<hostname># cp /mnt/data.<hostname>.CA.gz /opt/
<hostname># gzip -cd /opt/data.<hostname>.CA.gz | tar -xvf -
<hostname># cp /mnt/data.<hostname>.FSPTC.gz /opt/
<hostname># gzip -cd /opt/data.<hostname>.FSPTC.gz | tar -xvf -
<hostname># cp /mnt/data.<hostname>.FSAIN.gz /opt/
<hostname># gzip -cd /opt/data.<hostname>.FSAIN.gz | tar -xvf -
```
- For EMS/BDMS:**
- ```
<hostname># mount <NFS server ip>:/<shared directory> /mnt
<hostname># cp /mnt/oradata.<hostname>.gz /opt/
<hostname># cp /mnt/db.<hostname>.gz /opt/
<hostname># gzip -cd /opt/oradata.<hostname>.gz | tar -xvf -
```
- Step 4** Reboot the system.
- ```
<hostname># sync; sync;
<hostname># shutdown -y -g0 -i6
```

Step 5 Bring up Cisco BTS 10200 application.

```
<hostname># platform start
```

Step 6 Restore platform startup script.

```
<hostname># mv /etc/rc3.d/saved.S99platform /etc/rc3.d/S99platform
```

Step 7 Refer to Jumpstart documentation to set up disk mirroring.



Note

We recommend running this procedure during a maintenance window or when traffic is low.

Power Failure Recovery

One critical component of the Cisco BTS 10200 software is the memory resident database, also referred to as shared memory. Shared memory can be damaged by internal/external power supply failure.

The local status indicator for the shared memory database indicates that all actions needed to synchronize this database with the Active side while on standby have been completed. This status is tested when a platform starts up as Active. If the target state is Standby, the status indicator does not affect the normal startup sequence.

Power Fail Occurs Procedure

If power failure occurs, do the following:

1. Check the state of the surviving hosts and make sure that all platforms are still running.
2. Check the alarm logs if the EMS is available.

Power Is Restored Procedure

When power comes back on, the Cisco BTS 10200 software and all platforms should power up running in duplex active/standby.

-
- Step 1** Use the **nodestat** command to verify that all platforms are running with no failure indication.
- Step 2** If the platform shuts down or fails to come up, perform the following steps to determine the cause of the problem and determine the action to resolve it:
- a. Check the alarm logs to verify the system status.
 - b. Trace logs display the most significant events about the state of the platforms. Check and analyze the logs for details that may provide the cause of the failure.
-

Power Failure Scenarios

Four power failure scenarios are discussed in the following sections:

- [Power Failure on Single Host Computer, page 16-8](#)
Only one host of the two mated host computers is affected by the power outage.
- [Power Failure on Both Call Agent Computers, page 16-9](#)
Both mated host Call Agent computes are affected by the power outage.
- [Power Failure on Both Element Management System Computers, page 16-9](#)
Both mated host EMS computers are affected by the power outage,
- [Total System Power Outage, page 16-10](#)
All host computers are affected by the power outage.

Power Failure on Single Host Computer

If power failure occurs on one of the two sides while starting up a Standby platform, it can result in corrupted shared memory. The status indicator for the shared memory database will report “shared memory database in bad state” if the Standby platform is restarted as the Active platform.

Recovery Procedure

Depending on the current state of the mate to the host computer, the following procedure discusses alternatives on how to proceed:

No Failures on the Surviving Host

-
- Step 1** Verify that all platforms are running as Active.
 - Step 2** If all platforms are running as Active, restore power. Restoring power restarts all platforms running as Standby on the failing host.
-

Platform Failure on the Surviving Host

-
- Step 1** If power failure occurred while the surviving host computer is being brought up, restart the failing platform *immediately* (before power is restored on the other host).
 - Step 2** If the procedure does not work and reports a “bad state” for the shared memory, proceed to clear the shared data area and wait for the mate to be restarted.
 - Step 3** When power is restored, verify that all platforms are running by entering the **nodestat** command on the recovering host computer.
 - Step 4** Restart the failing platform on the host computer that is not affected by the power outage. The platform should come up as Standby.
-

Platform Failure Not Due To “Bad State” of the Shared Memory on the Mate Host Computer or Any Failure on the Recovering Host Computer

-
- Step 1** Check the alarm logs and search for alarms belonging to the failing platforms.
 - Step 2** The trace logs display the most significant events about the state of the platforms. Check and analyze the logs for details that may provide the cause of the failure. If possible, fix the problem.
 - a.** If the system can run in simplex, send the logs to Cisco Technical Assistance Center (TAC) for diagnosis and assistance.
 - b.** If the system cannot run simplex, run the procedure for a duplex power failure.
-

Power Failure on Both Call Agent Computers

-
- Step 1** If any platforms start, take them down first.
- Step 2** Clear data directories on both sides and perform a fresh download from the EMS, as shown in the following steps. Do the following on all platforms:
- Enter the following command:

```
cd <platform>/bin/data; rm *
```
 - Restart both sides using the following command:

```
platform start all
```
 - Do a fresh download (extract Oracle data from the EMS and send it to the Call Agent). See the [Cisco BTS 10200 Softswitch CLI Database](#) for the commands.
 - Check transaction queue—make sure data is going from the EMS to the CA.
 - Enter the command **audit db ems** to make sure everything is in sync.
- Step 3** Fix discrepancies by means of CLI commands.

**Caution**

This may take hours to complete and during this time, call processing is lost; that is why it is critical that there be no single point of failure in the power feeds.

Power Failure on Both Element Management System Computers

-
- Step 1** If the platforms start, shut them down first.
- Step 2** Audit the Oracle database.
- Step 3** Check the mysql database.
- Step 4** Restart both sides using the following command:

```
platform start all
```
- Step 5** Enter the command **audit db ems** to make sure everything is in sync.
- Step 6** Fix discrepancies using CLI commands.

**Caution**

This may take hours to complete and during this time, call processing is lost; that is why it is critical that there be no single point of failure in the power feeds.

Total System Power Outage

- Step 1** If the platforms start, take them down first.
- Step 2** Audit the Oracle database.
- Step 3** Check the mysql database.
- Step 4** Clear the data directories on both call agent sides and do a fresh download from the EMS.
- Step 5** On all platforms repeat the following steps:
- Enter the following command:

```
cd <platform>/bin/data; rm *
```
 - Restart both sides using the following command:

```
platform start all
```
 - Do a fresh download (extract Oracle data from the EMS and send it to the Call Agent). See the [Cisco BTS 10200 Softswitch CLI Database](#).
 - Check the transaction queue to make sure that data is going from the EMS to the CA.
 - Enter the command **audit db ems** to either make sure everything is in sync.
- Step 6** Fix discrepancies using CLI commands.

**Caution**

This may take hours to complete and during this time, call processing is lost; that is why it is critical that there be no single point of failure in the power feeds.

Element Management System Database Recovery From Hot Back Up

This section provides procedures you can use to restore your Oracle EMS database data files from the most current hot back up and then recover your database from the back up. If additional archive log back up (by ora_arch_backup.ksh) was done after the hot back up, the additional archive log back up file sets need to be restored also. All of these back up file sets are assumed to be located on the remote FTP site.

Directory to restore back up files: /opt/oraback.

The following assumptions were made for this procedure:

Daily back up schedule:

2:00 AM—daily hot back up (by ora_hot_backup.ksh process)

18:00 PM—daily archive log back up (by ora_arch_backup.ksh process)

Oracle databases on both primary and secondary EMS systems crashed completely at January 10, 2007, 20:00pm.

Recovery Goal

The goal in the scenario above is to recover the primary EMS Oracle database by using your most recent back ups.

In this case, since the database crashed January 10, 2007, 20:00pm, the back up file sets with timestamp '200701100200' from 2:00am hot back up and those with timestamp '200701101800' from 18:00 archive log back up must be restored. Timestamp is formatted as YYYY:MM:DD:hh:mm.

If your database crashes before the archive log back up, you need to restore only the 2:00 am hot back up file sets.

If your system does not perform extra archive log back up daily by ora_arch_backup.ksh, use back up file sets from hot back up only.

In this sample scenario, the primary EMS database is recovered first to resume operation. Then the secondary EMS will be recovered using the procedures that recover data from the primary EMS.

**Note**

Before this recovery process is applied, it is assumed that the entire system, including all corrupted applications, has been restored.

Recovering the Primary Element Management System Database

Perform the following procedure on the primary EMS system to recover the primary EMS database from your most recent back up files:

-
- Step 1** Make sure the platform is shut down and the system cron process has stopped.
 - Step 2** Log in as **root**.

Step 3 Enter the following commands to shut down the system:

```
platform stop all

svcadm disable svc:/system/cron
```



Note Execute **platform stop all** and **stop_cron.sh** on the secondary EMS also if the secondary EMS platform is active.

Step 4 Log in as **oracle** user, or **su – oracle**.

Step 5 Enter the following command to verify that there is enough free disk space:

```
df -k /opt/oraback
```

The EMS system must have enough disk space in the /opt/oraback directory to restore all database data files and archive log files. The database data files can take up to 3.6 gigabits (GB) if fully populated with data; each archive log file requires 5 MB additional space. The number of archive log files in the back up set can be identified from the optical1_ora_hot_full_backup_<timestamp>.log and/or the optical1_ora_arc_incr_backup_<timestamp>.log file in /opt/oraback directory.

Step 6 Restore targeted back up file sets from the remote FTP site.

FTP the targeted database back up file sets from the remote FTP server to the /opt/oraback directory on the EMS system. Then uncompress all the “.Z” files.

a. Enter the following commands:

```
cd /opt/oraback
ftp <remote_ftp_server>
```

b. Log in as **oracle**.

c. Enter the password (default password is *ora00*).

d. Enter the following commands:

```
ftp cd <remote_backup_directory>
ftp bin (* Use binary transfer mode *)
```

e. Get the following files. If archive log back up is not performed, get only the hot back up files.

Back up files from 2:00 hot back up:

- optical1_arc_full_1_167:200701100200.Z
- optical1_arc_full_1_168:200701100200.Z
- optical1_ctl_binary: 200701100200.Z
- optical1_ctltrc:200701100200:tar.Z
- optical1_hot_full_1_166:200701100200.Z
- optical1_ora_hot_full_backup_200701100200.log

Back up files from the 18:00 archive log back up:

- optical1_arc_incr_1_169:200701101800.Z
- optical1_ctl_binary:200701101800.Z
- optical1_ctltrc:200701101800:tar.Z
- optical1_ora_arc_incr_backup_200701101800.log

```
ftp> prompt
```

```
ftp> mget optical1*200701100200*
ftp> mget optical1*200701101800*
ftp> quit
ls *200701100200*
ls *200701101800*
```

f. Uncompress your files:

```
uncompress *200701100200*.Z
uncompress *200701101800*.Z
```



Note

At this point all files are restored from remote ftp server in the `/opt/oraback` directory. You are now ready to apply the database recovery processes to bring your database up to the point of your last back up.

Step 7 Clean up old database data files by entering the following commands:

```
cd /data1/oradata/optical1
```



Note

If you are on the secondary EMS, cd to `/data1/oradata/optical2`.

```
rm data/* db1/* db2/* index/*
```

```
df -k /data1/oradata
```



Note

You must have a minimum of 3.6 GB free disk space on `/data1/oradata/optical1` to accommodate all database data files from back up.

Step 8 Restore the back up binary control file to the database target directories:

Use the most current back up binary control file. In this case use the `optical1_ctl_binary:200701101800` file from 18:00pm archive log back up. If archivelog back up was not restored use the binary control file from 2:00am back up. Copy the back up binary control file to both `db1/control01.dbf` and `db2/control02.dbf` files.

```
cp /opt/oraback/optical1_ctl_binary:200701101800 db1/control01.ctl
cp /opt/oraback/optical1_ctl_binary:200701101800 db2/control02.ctl
```

Step 9 Recover the database using the `recover_db_until_time.ksh` script.

The `recover_db_until_time.ksh` script uses the restored binary control file to mount the database, restores all data files from the restored database data-sets, applies all applicable archivelog files through the restored archivelog file sets, then finally opens the database with the reset logs option and adds the temp file back up to temp tablespace. When this script is completed successfully, database is recovered to the point of time of the back ups.

Before executing the `recover_db_until_time.ksh`, shut down all Oracle instance processes.

```
cd /opt/oracle/admin/backup
./recover_db_until_time.ksh $ORACLE_SID
```

System response similar to the following is displayed:

```

*****
This process will perform database recovery using RMAN backup datasets.

Target: hostname=priems16 database=optical1

You must complete the following procedures before this process:

1. platform stop all
2. stop_cron.sh
3. restore all required backup datasets to /opt/oraback directory
4. copy optical1_ctl_binary file to /data1/./<db1 and db2>

*****

Do you want to continue? [y/n] y << Enter y

Log file: /opt/oracle/tmp/recover_db_until_time_200701101636.log

<Thu Jan 10 16:36:51 CST 2007> ./recover_db_until_time.ksh started.
Mounting control file...
Connected to an idle instance.
ORACLE instance started.

Total System Global Area 287912096 bytes
Fixed Size 73888 bytes
Variable Size 181915648 bytes
Database Buffers 104857600 bytes
Redo Buffers 1064960 bytes
Database mounted.

Restoring all datafiles ..
RMAN> 2> 3> 4> 5> 6> 7> 8>
<Thu Jan 10 16:40:15 CST 2007> All datafiles are restored.

<Thu Jan 10 16:40:15 CST 2007> Begin to recover database.

Recover database until time '20070111 14:00:13' << until time is always the restored
timestamp+1day
Last logseq=6782 thread=1

RMAN msglog file: /opt/oracle/tmp/recover_db_until_time_200701101636.log
RMAN> 2> 3> 4> 5> 6> 7> 8>
**** You can Ignore RMAN error messages regarding to: << Ignore this error message from
the log file
**** MAN-08060: unable to find archivelog
**** RMAN-08510: archivelog thread=1 sequence=6783
**** .....
**** RMAN-06054: media recovery requesting unknown log:

<Thu Jan 10 16:44:27 CST 2007> Database recovery ended.

<Thu Jan 10 16:44:27 CST 2007> Alter database open resetlogs
Connected.
Database altered.
...
Database is successfully recovered.

<Thu Jan 10 16:44:38 CST 2007> ./recover_db_until_time.ksh ended.

```

Post Recovery–Cold Back Up

Once you have recovered your database, you need to make a cold back up of the database using the **dbadm -E cold_backup** command. The following tar files are created from the cold back up script. You need to save a copy of these files to the `/opt/oraback` directory. Make sure that the following files are saved to the off site FTP server:

- `/opt/oracle/tmp/optical1_DB_upd.tar.gz`
- `/opt/oracle/tmp/optical1_ADMIN_upd.tar`
- `/opt/oracle/tmp/optical1_upd.crontab`

Step 1 Log in as **oracle**, or **su - oracle**:

Step 2 Enter the following command:

```
dbadm -E cold_backup
```



Note This process can take more than 10 minutes to complete, depending on the volume of data in the database.

Text similar to the following is displayed:

This process performs the following tasks:

1. Shutdown optical1 database on priems09.
2. Backup `/opt/oracle/admin` directory (except arch dump and log).
3. Cold backup database.
4. Backup oracle crontab file.
5. Startup database.

The following backup files are generated at the end of process:

```
/opt/oracle/tmp/optical1_DB_upd.tar.gz
/opt/oracle/tmp/optical1_ADMIN_upd.tar
/opt/oracle/tmp/optical1_upd.crontab
```

```
Free disk space left on /opt/oracle/tmp: 1383 MB
```

```
*****
```

```
LOG file: /opt/oracle/tmp/ora_cold_backup.log
```

```
Do you want to continue? [y/n] y
```

Step 3 Once the cold back up is completed, save a copy of the back up files to the `/opt/oraback` directory for the ftp script to transfer off site.

```
cd /opt/oracle/tmp
cp optical1_ADMIN_upd.tar /opt/oraback
cp optical1_upd.crontab /opt/oraback
cp optical1_DB_upd.tar.gz /opt/oraback
```

Step 4 Clean up the restored files in */opt/oraback* directory to claim the disk space back.

```
ls /opt/oraback/*200701100200*
ls /opt/oraback/*200701101800*

rm /opt/oraback/*200701100200*
rm /opt/oraback/*200701101800*
```

Step 5 Resume operations.

You are now ready to shut down the Oracle database and start the platform and cron process.

Step 6 Log in as **root** or **su - root**

Step 7 Enter the following commands:

```
su -root
platform stop -i oracle
platform start
svcadm enable svc:/system/cron
nodestat
```

The recovery of the primary EMS database is now complete. To recover the secondary EMS database, copy data from the primary EMS database. Refer to the [“Recovering the Element Management System Database From Another Database”](#) section on page 16-17.

Recovering the Element Management System Database From Another Database

This section provides the procedures to recover one corrupted EMS database from another active database.

Recovery Procedures

The steps in this section show you how to recover a corrupted EMS database from the other active peer database assuming one of the two following scenarios (this procedure applies to both scenarios):

- Scenario 1 The primary EMS database is corrupted. You would like to restore data from the secondary EMS database.
- Scenario 2 The secondary EMS database is corrupted. You would like to restore data from the primary EMS database.

Step 1 On the active EMS site, terminate the DBHeartBeat process and disable push job (job 2).

- a. On the active EMS site, log in as **oracle**, or **su - oracle**.
- b. Enter the following command to terminate DBHeartBeat process:

```
$ dbinit -H -i stop  
  
$ ps -ef | grep hbmgr | grep -v grep
```

- c. Disable push job (job 2).

```
$ dbadm -A disable_push_job
```

- d. Respond **y** at the prompt and enter the following:

```
$ dbadm -r get_broken_jobs
```

Text similar to the following is displayed:

```
2 Y 0 declare rc binary_integer; begin rc := sys.dbms_defer_s  
ys.push(destination=>'OPTICAL1', stop_on_error=>FALSE,  
delay_seconds=>0, parallelism=>1); end;
```

Step 2 Shut down all processes on the corrupted EMS site:

- a. On the corrupted EMS site, log in as **root**.

- b. Stop the cron process and shut down the platform:

```
# svcadm disable svc:/system/cron
# platform stop all
# nodestat
```

Verify whether all database processes are terminated:

```
# nodestat
# ps -ef | grep ora_
# ps -ef | grep hbmgr
# ps -ef | grep tnslsnr
```



Tip You can use `kill -9` to kill any process not being terminated by `platform stop all`.

```
# ipcs -p | grep oracle
```



Tip You can use the `ipcrm` command to remove any shared memory or semaphore still allocated to `oracle` now. For example: `ipcrm -m <identification (ID)>`, `ipcrm -s <ID>`

Step 3 **This step is optional.** Save all current database logs and trace files on the corrupted EMS site.

If the disk that stores the Oracle database dump and log files still exists, you can save the dump and log files to use later if needed.

- a. On the corrupted EMS site, log in as **oracle**, or **su - oracle**:
- b. Enter the following commands:

```
$ su - oracle
$ cd /data1/dump
$ tar -cvf /opt/oraback/data1_dump_corrupted.tar *
$ cd /opt/oracle/tmp
$ tar -cvf /opt/oraback/opt_oracle_tmp_corrupted.tar *
```



Timesaver

You can `gzip` or compress the tar files if they are very large.

Step 4 On the corrupted EMS site, rebuild the Oracle database from one of the following three options:

- Option 1 If only database is corrupted and the Cisco BTS 10200 reinstallation is not required, go to [Step 5](#) to reload the database from the database back up file. Continue to [Step 6](#).
- Option 2 If the entire system is corrupted and flash archive system back up is available, recover the system from the flash archive, as detailed in the [“Flash Archive Restore” section on page 16-2](#). The flash archive back up file should have the Cisco BTS 10200 applications included. Continue to [Step 6](#).
- Option 3 If the entire system is corrupted and the flash archive back up file is not available, you must jump start the system, and reinstall the Cisco BTS 10200 software from the installation CD, as shown below, [“Reinstalling the Cisco BTS 10200 Software on the Corrupted EMS” section on page 16-19](#).

Reinstalling the Cisco BTS 10200 Software on the Corrupted EMS

- a. Update /etc/opticall.cfg file. You can copy this file from active EMS. Verify that the contents are correct.
- b. Create the /opt/ems/utills directory, if it does not already exist. Enter the following command:


```
# mkdir -p /opt/ems/utills
```
- c. FTP the file /opt/ems/utills/Version from the active EMS to the corrupted EMS, then rename the file to version.save for reference. Enter the following commands:


```
# cd /opt/ems/utills
# cat Version.save
900-xx.yy.zz.VVV
```
- d. Enter the following commands to create the version file from version.save, but change the version number to D00 (D zero zero). This D00 version value is only a tag; it does not affect the target version to be installed.


```
# sed 's/...$/D00/' Version.save > Version
# cat Version
900-xx.yy.zz.D00
```
- e. Change to the CD Build directory and run **install.sh** with the -upgrade option. Enter the following command:


```
# ./install.sh -upgrade
```



Note For procedures on how to mount the installation CD and load or untar the software packages to **/opt/Build**, see the following sections in “*Application Installation Procedure (Release 4.4)*”:

- “Load the K9-opticall.tar(.gz) File on the EMS and CA/FS Platforms” on page 15
- “Load the K9-oracle.tar(.gz) File on the EMS” on page 23

- f. After the corrupted EMS system is reinstalled, enter the commands below to shut down the platform and only start up Oracle listener and database:


```
# platform stop all
# su - oracle
# dbinit -L -E -i start
```

Continue to [Step 6](#).

Step 5 Reload the database from cold back up to the corrupted EMS site.

If the EMS system is intact, but only the Oracle database is corrupted, you can use the cold back up tar file to restore the database data files. The cold back up tar file `optical1_DB_upd.tar.gz` is for the primary EMS, and `optical2_DB_upd.tar.gz` is for the secondary EMS.

- If the tar file is not in /opt/oraback directory and the same file still exists in the /opt/oracle/tmp directory, copy this file from /opt/oracle/tmp to /opt/oraback directory.
- If the file does not exist on either directory, restore this file from remote FTP server to /opt/oraback directory, then execute the steps in this section to restore database data files from the cold back up tar file.



Note If there is no cold back up database tar file, you can restore database from hot back up. Refer to the “[Element Management System Database Recovery From Hot Back Up](#)” section on [page 16-11](#) to recover your database from hot back up.

- a. Restore the database from the cold back up tar file. Log in as oracle:

If the corrupted database is the primary EMS database use the optical1_DB_upd.tar.gz file:

```
$ cd /data1/oradata/optical1
$ rm -r data/* db1/* db2/* index/*
$ gzip -cd /opt/oraback/optical1_DB_upd.tar.gz | tar xvf -
```

If the corrupted database is the secondary EMS database, use the optical2_DB_upd.tar.gz file:

```
$ cd /data1/oradata/optical2
$ rm -r data/* db1/* db2/* index/*
$ gzip -cd /opt/oraback/optical2_DB_upd.tar.gz | tar xvf -
```

- b. Start the database restore process. After the database data files are restored, execute the following command to start up the EMS database process:

```
$ dbinit -L -E -i start
```

Step 6 Stop all transactions except northbound traffic on the active EMS. From the active EMS side, stop all transactions to the database except northbound traffic and status control update from CA or FS.



Caution There is no CLI provisioning and Simple Network Management Protocol (SNMP) processes must be stopped.

- a. Log in as root.
b. Enter the following command:

```
# pkill smg3
```

Step 7 Copy data from the active EMS database to the corrupted database:



Caution During this step the **dbadm -A copy_all** process will truncate local tables first, then copy data from the tables on the other site. Make sure that you execute this step on the corrupted EMS side only.

- a. On the corrupted EMS side, log in as **oracle**, or **su – oracle**.



Note Make sure that you are on the corrupted database site.

- b. Enter the following command:

```
$ dbadm -A copy_all
```

Text similar to the following is displayed:

```
*****
You are about to execute the following process:

==> Copy all OAMP/OPTICALL/BILLING tables from remote DB optical1 at priems47

database: optical2
hostname: secems47

*****
```

- c. At the prompt, enter y to continue:

```
Do you want to continue? [y/n] y
```

Text similar to the following is displayed:

```
***This will EMPTY all the tables on:
*** local host ==> secems47
*** local database ==> optical2
***
*** Then copy data from remote DB optical1 at priems47
```

- d. At the prompt, enter y to continue:

```
Do you want to continue? [y/n] y
```



Note This process will take some time. At a database with maximum capacity, it can take approximately 2 hours to copy all operations, administration, maintenance and provisioning (OAMP) and OPTICALL tables.

Response similar to the following example is displayed:

```
<Mon Jan 24 11:40:23 CST 2005> INFO: DMMgr::Configuration loaded
<Mon Jan 24 11:40:23 CST 2005> INFO: DMMgr::243 rows updated
<Mon Jan 24 11:40:24 CST 2005> INFO: DMMgr::Disabling Foreign Key constraints for
BILLING.
<Mon Jan 24 11:40:24 CST 2005> INFO: DMMgr::Disabling triggers for BILLING...
<Mon Jan 24 11:40:25 CST 2005> INFO: copy table => BILLING.BILLING_ACCT_ADDR..
<Mon Jan 24 11:40:26 CST 2005> INFO: copy table => BILLING.BILLING_ACCT_ADDR ...
...
Mon Jan 24 11:40:28 CST 2005> INFO: copy tables => OK=3, FAIL=0, SKIP=0, OTHERS=0
<Mon Jan 24 11:40:28 CST 2005> INFO: DMMgr::Enabling Foreign Key constraints for
BILLING.
<Mon Jan 24 11:40:28 CST 2005> INFO: DMMgr::Enabling triggers for BILLING...
<Mon Jan 24 11:40:29 CST 2005> INFO: DMMgr::Disabling Foreign Key constraints fo
r OAMP...
<Mon Jan 24 11:40:29 CST 2005> INFO: DMMgr::Disabling triggers for OAMP...
<Mon Jan 24 11:40:29 CST 2005> INFO: copy table => OAMP.CALL_TRACE..
<Mon Jan 24 11:40:30 CST 2005> INFO: copy table => OAMP.CALL_TRACE ...OK(0 row)
...
<Mon Jan 24 11:41:41 CST 2005> INFO: copy tables => OK=50, FAIL=0, SKIP=0, OTHER
S=0
<Mon Jan 24 11:41:41 CST 2005> INFO: DMMgr::Enabling Foreign Key constraints for
OAMP...
<Mon Jan 24 11:41:41 CST 2005> INFO: DMMgr::Enabling triggers for OAMP...
<Mon Jan 24 11:41:41 CST 2005> INFO: DMMgr::Disabling Foreign Key constraints for
OPTICALL...
```

```

<Mon Jan 24 11:42:07 CST 2005> INFO: DMMgr::Disabling triggers for OPTICALL...
<Mon Jan 24 11:42:47 CST 2005> INFO: copy table => OPTICALL.AAA_SERVER_GRP..
<Mon Jan 24 11:42:48 CST 2005> INFO: copy table => OPTICALL.AAA_SERVER_GRP ...OK (0
row)
...
Mon Jan 24 11:46:41 CST 2005> INFO: copy table => OPTICALL.WIRETAP..
<Mon Jan 24 11:46:42 CST 2005> INFO: copy table => OPTICALL.WIRETAP ...OK(0 row)
<Mon Jan 24 11:46:42 CST 2005> INFO: copy tables => OK=190, FAIL=0, SKIP=0, OTHERS=0
<Mon Jan 24 11:46:42 CST 2005> INFO: DMMgr::Enabling Foreign Key constraints for
OPTICALL...
<Mon Jan 24 11:47:20 CST 2005> INFO: DMMgr::Enabling triggers for OPTICALL...

```

- Step 8** On the active EMS, truncate all replication queues since all data are already copied over, then start the DBHeartBeat process. The DBHeartBeat process automatically enables the broken push job.

From active EMS side:

- a.** Truncate replication queues. Enter the following commands:

```

# su - oracle

$ dbadm -A truncate_def

$ dbadm -r get_unpushed_trans

```

Text similar to the following is displayed:

```
no rows selected
```

- b.** Start the DBHeartBeat process. Enter the following command:

```

$ dbinit -H -i start

$ dbadm -r get_broken_jobs

```

Text similar to the following is displayed:

```

2 N          0          declare rc binary_integer; begin rc :=
sys.dbms_defer_sys.push(destination=>'OPTICAL1',
stop_on_error=>FALSE,delay_seconds=>0, parallelism=>1); end;

```

- Step 9** Verify the database status and audit contents of tables. This step can be executed on either the primary or secondary EMS site. In this case, it is executed on the primary EMS site.

- a.** On the active EMS site, log in as **oracle**, or **su - oracle**.

- b.** Enter the following command:

```
$ dbadm -C db
```

Response similar to the following is displayed:

```
OPTICAL1::Deftrandest is empty?          YES
OPTICAL1::dba_repcatlog is empty?       YES
OPTICAL1::Deferror is empty?           YES
OPTICAL1::Deftran is empty?            YES
OPTICAL1::Has no broken job?           YES
OPTICAL1::JQ Lock is empty?            YES

OPTICAL2::Deftrandest is empty?          YES
OPTICAL2::dba_repcatlog is empty?       YES
OPTICAL2::Deferror is empty?           YES
OPTICAL2::Deftran is empty?            YES
OPTICAL2::Has no broken job?           YES
OPTICAL2::JQ Lock is empty?            YES
```

```
Checking table => OPTICALL.AGGR...OK
Checking table => OPTICALL.ANI...OK
..
..

Number of tables to be checked: xxx
Number of tables checked OK: xxx
Number of tables out-of-sync: 0
```

Step 10 Synchronize table contents from the uncorrupted EMS site to the corrupted EMS site.

If the **dbadm -C db** command from [Step 9](#) returns out-of-sync table(s) like the examples below, follow the commands in [Step a.](#) and [Step b.](#) (below) to synchronize the contents of data from the active EMS database to the corrupted EMS database.

Text similar to the following example is displayed:

```
Number of tables to be checked: 130
Number of tables checked OK: 127
Number of tables out-of-sync: 3
```

List of out of sync tables:

```
OAMP.TABLE_NAME => 22/0
```

In this example, one table owned by the OAMP is out of sync. Follow the steps below to synchronize the contents of the tables:



Note

Execute these commands on the corrupted EMS database to synchronize the table.

Truncate the content of the table on the local database, then copy the data from the remote database:

- a. Log in as oracle, or su - oracle:
- b. Enter the following commands:


```
$ dbadm -A copy -o <owner> -t <table_name>
```

```
$ dbadm -A copy -o oamp -t table_name
```

Step 11 Verify the Oracle crontab file:

- a. Log in as oracle, or su - oracle:
- b. Verify the Oracle crontab file on the corrupted EMS site. Compare the schedules of jobs with those on the active EMS site. If any schedule needs to be modified, enter the following command:


```
$ crontab -e
```

Step 12 Shut down the Oracle database and start up the platform on the standby EMS sites. Both primary and secondary databases have identical data. You must start the platform and system cron processes on the standby EMS.

On the corrupted EMS site, log in as root and enter the following commands to bring up the platform:

```
# su - oracle
$ dbstat -a -f
$ dbstat -j bts10200_bts_stat_daily -J enable -f
$ dbadm -s get_dbms_schedules | grep -i stat_daily | grep -i gather_bts
    BTS10200_BTS_STAT_DAILY BTS10200_GATHER_BTS SCHEDULED TRUE
$ exit
# platform stop -i oracle
# platform start
# svcadm enable svc:/system/cron

# su - oracle

$ dbadm -A stat_bts_job
```

The EMS database recovery from another database is now complete.

Fresh Download

The **Fresh Download** command refreshes data in Call Agent shared memory. It recovers the data in the Call Agent shared memory in the event that shared memory data cannot be recovered by any other means. The fresh download wipes out Call Agent shared memory data and causes a total outage.

**Caution**

Do not use this command on **any** live traffic production systems. Please contact Cisco TAC regarding the use of this command for disaster recovery.

You can perform the command by ID. The **download by ID** command allows you to copy database information from the EMS to a specific CA or FS. If a CA ID is not specified, the command copies to all IDs.

Use one of the following examples to perform a fresh download by ID.

```
download database target=ca; id=CA146
download database target=fsptc; id=FSPTC135
download database target=fsain; id=FSAIN125
```

See the [Cisco BTS 10200 Softswitch CLI Database](#) for table and token descriptions.

Call Agent Database Download and Recovery

This call agent database download and recovery procedure is recommended as a last resort for recovering corrupted call agent databases.


Note

Please contact Cisco TAC regarding the use of this procedure for call agent database recovery.


Note

If transactions are stuck in the queue, execute a **delete transaction-queue** CLI command before beginning this procedure.

Step 1 Execute the following **download database** CLI commands.

```
CLI> download database target=ca; file=/opt/tmp/download-CA
CLI> download database target=fsain; file=/opt/tmp/download-FSAIN
CLI> download database target=fsptc; file=/opt/tmp/download-FSPTC
```

Step 2 Perform a **platform stop all** command on both call agents.

Step 3 Delete the following data directories on the primary and secondary call agents.

```
cd /opt/OptiCall
rm -rf */bin/data
```

Step 4 Start the primary call agents or feature servers with the **platform start** command.

Step 5 FTP the files created in Step 1 to the primary call agent.

```
download-CA -ftp to-> /opt/OptiCall/CAxxx/bin
download-FSPTC -ftp to-> /opt/OptiCall/FSPTCxxx/bin
download-FSAIN205 -ftp to-> /opt/OptiCall/FSAINxxx/bin
```

Step 6 Go into mysql on the primary call agent or feature server and upload the database as shown. This is to be done in parallel with three different sessions opened to the appropriate call agent or feature server.

a. From the /opt/OptiCall/CAxxx/bin command line:

```
./dbm_sql.CAxxx ./data ./catalog < download-CA
```

b. From the /opt/OptiCall/FSPTCxxx/bin command line:

```
./dbm_sql.FSPTCxxx ./data ./catalog < download-FSPTC
```

c. From /opt/OptiCall/FSAINxxx/bin command line:

```
./dbm_sql.FSAINxxx ./data ./catalog < download-FSAIN
```

Step 7 Start the secondary call agents or feature servers with the **platform start** command.

Step 8 Control all provisioned network devices to in service using manually generated return to service scripts.



Note The following steps should be run in parallel to limit the amount of down time; for example, one engineer working on returning subscribers to in service and one engineer working on returning trunks, trunk-grps, and so forth to in service.

a. Complete the following items with scripts manually created through CLI.

```
- control trunks oos
- control trunk-grps oos
- control h323gws oos
- unequip trunk terminations
- control h323gws ins
- control trunk-grps ins
- equip trunk terminations
- control trunks ins
```

b. Bring subscribers into service utilizing the cs-control tool. A script is generated with all the subscriber terminations. An example of the script is as follows:

```
control subscriber-termination id=sub1; mode=forced; target-state=INS
control subscriber-termination id=sub2; mode=forced; target-state=INS
control subscriber-termination id=sub3; mode=forced; target-state=INS
control subscriber-termination id=sub4; mode=forced; target-state=INS
```

c. Place the completed script in the /opt/OptiCall/CAxxx/bin directory on the primary call agent.

d. Invoke the cs-control script from /opt/OptiCall/CAxxx/bin directory to place subscribers in service.

```
cs-control data <name of script from step 8b>
```

Recovering Shared Memory Data

The **download database** command refreshes data in the Call Agent (CA) shared memory. In the event that shared memory data cannot be recovered, the command recovers data in the CA shared memory. The **download database** command wipes out Call Agent shared memory data and causes a total outage.



Caution

Read this section in its entirety before attempting this procedure.

Do not use this command on any live traffic production systems. Contact Cisco TAC regarding the use of this command for disaster recovery.



Caution

Do not download the database through the console port because the TTY can cause long delays.

Downloads of the CA, FSAIN and FSPTC applications can be done in parallel.

You can perform the **download database** command by ID. The **download by ID** command allows you to copy database information from the EMS to a specific CA or Feature Server (FS). If a CA ID is not specified, the command copies to all IDs.

Use one of the following examples to download database by ID.

```
download database target=ca; id=CAxxx
download database target=fsptc; id=FSPTCxxx
download database target= fsain; id=FSAINxxx
```

See the [Cisco BTS 10200 Softswitch CLI Database](#) for table and token descriptions.

Recovering Shared Memory

Perform the following procedure:

Step 1 Check the transaction queue. Use the following command:

```
show transaction-queue
```

Step 2 If you find any transaction in the queue, delete the queue by entering the following CLI command.



Note

There should be no provisioning activity on the system—if the **show transaction-queue** command does not return the message “Void of entries,” assume that a transaction is in the queue. If there is no transaction in the queue, proceed to [Step 7](#).

```
delete transaction-queue target=CAxxx; transaction-id=<id>
```

The **transaction-id** parameter enables you to delete only one transaction at a time from the `transaction_queue` table. Note that to delete an entry from the `transaction-queue`, you should login as **ciscouser**.

Examples:

```
delete transaction-queue target=CAxxx; transaction-id=<id>
delete transaction-queue target=FSPTCxxx; transaction-id=<id>
delete transaction-queue target=FSAINxxx transaction-id=<id>
```

If there are thousands of entries stuck in the transaction queue, it is recommended to flush all the entries from Oracle after logging in as **oamp** user.

Perform the following steps to log in as **oamp** user and to flush all the entries from Oracle:

-
- Step 1** <hostname># **su - oracle**
- Step 2** <hostname>\$ **sqlplus oamp/oamp**
- Step 3** SQL> **delete from transaction_queue;**
- Step 4** SQL> **commit;**
- Step 5** SQL> **exit**
- Step 6** <hostname>\$ **exit**
-

In the following example, 138327 transactions are stuck in the queue, and the display of the transactions (using the **show** command) is limited to 2. This example shows that when many transactions are stuck in the queue, deleting one transaction at a time takes huge amount of time. In such scenarios, it is recommended to flush all the entries from Oracle.

```
CLI> show transaction_queue limit=2
```

```
TRANSACTION_ID=1242510999092
SEQUENCE_NUM=0
TARGET=CA101
STATEMENT=delete from MGW_PROFILE;
TIMESTAMP=2009-05-16 17:26:39
ACTIVE_TARGET=Y
USERNAME=null
TERMINAL=null
STATUS=FAILED
```

```
TRANSACTION_ID=1242510999092
SEQUENCE_NUM=1
TARGET=CA101
STATEMENT=delete from NDC;
TIMESTAMP=2009-05-16 17:26:39
ACTIVE_TARGET=Y
USERNAME=null
TERMINAL=null
STATUS=PENDING
```

```
Reply : Success: Entries 1-2 of 138327 returned.
```

If you want to delete the first transaction, enter the following command:

```
CLI> delete transaction_queue target=CA101;transaction_id=1242510999092;
Reply : Success: CLI delete successfully
```

- Step 7** Enter the **shared memory** command to recover the CA and FS databases. Enter following CLI download database commands:

```
download database target=CA; file=/tmp/download-CA
```

```
download database target=FSAIN; file=/tmp/download-FSAIN
download database target=FSPTC; file=/tmp/download-FSPTC
```

Step 8 Stop both CA platforms. Enter the following command:

```
platform stop all
```

Step 9 Delete the following data directories (as shown below) on the primary and secondary CA and FSs. Enter the following commands:

```
cd /opt/OptiCall
\rm -rf */bin/data
```

Step 10 Use ftp to transfer the file created in Step 3 to the primary CA. Place the files in the following directories:

```
download-CA in /opt/OptiCall/CAxxx/bin
download-FSPTC in /opt/OptiCall/FSPTCxxx/bin
download-FSAIN in /opt/OptiCall/FSAINxxx/bin
```

Step 11 Start the primary CA and FS platforms. Enter the following command:

```
platform start
```

Step 12 Upload the database files using the dbm_dql tool as shown below.



Note Perform this step in parallel with three different sessions opened to the appropriate CA or FS.

The following steps may take up to 2 hours, depending on the size of the database being recovered.

a. Enter the following from the /opt/OptiCall/CAxxx/bin command line:

```
/dbm_sql.CAxxx ./data ./catalog < download-CA
```

b. Enter the following from /opt/OptiCall/FSPTCxxx/bin command line:

```
/dbm_sql.FSPTCxxx ./data ./catalog < download-FSPTC
```

c. Enter the following from /opt/OptiCall/FSAINxxx/bin command line:

```
/dbm_sql.FSAINxxx ./data ./catalog < download-FSAIN
```

Step 13 Start the secondary CA and FS platforms after [Step 12](#) completes.

```
platform start
```

Step 14 Continue to the following procedure, [“Restoring Subscriber and Trunk Terminations to Service”](#) section on page 16-31.

Restoring Subscriber and Trunk Terminations to Service

Control all provisioned network devices in service, as shown in the following subsections.



Caution

You must run the following procedures in parallel, to limit the amount of down time. For example, one engineer can be working on subscribers while another engineer is working on trunks or trunk groups.

Controlling Trunks and Trunk Groups

Control trunks and trunk groups with scripts created through the CLI commands listed in [Table 16-1](#).

Table 16-1 Controlling Trunks and Trunk Groups Examples

CLI Command	Example
control trunks oos	control trunk-termination tgn-id=1;cic=1-24;mode=forced; target-state=oos;
control trunk-grps oos	control trunk-grp id=1;mode=forced;target-state=oos;
control h323gws oos	control h323-gw id=h323;mode=forced;target-state=oos;
unequip trunk terminations	unequip trunk-termination tgn-id=1;cic=all;
control h323gws ins	control h323-gw id=h323;mode=forced;target-state=ins;
control trunk-grps ins	control trunk-grp id=1;mode=forced;target-state=ins;
equip trunk terminations	equip trunk-termination tgn-id=1;cic=all;
control trunks ins	control trunk-termination tgn-id=1;cic=1-24;mode=forced;target-state=ins;

Using the cs-control Tool to Bring Subscribers In-Service

Use the cs-control tool to bring subscribers in-service. Obtain the cs-control tool from Cisco. You must write a script (as shown in the following example) to be used by the cs-control tool.

Example of a Script

```
control subscriber-termination id=sub1; mode=forced; target-state=INS
control subscriber-termination id=sub2; mode=forced; target-state=INS
control subscriber-termination id=sub3; mode=forced; target-state=INS
control subscriber-termination id=sub4; mode=forced; target-state=INS
```



Note The completed script is placed in the /opt/OptiCall/CAxxx/bin directory on the primary CA.

The cs-control script is invoked from the /opt/OptiCall/CAxxx/bin directory to place subscribers in-service as follows:

```
cs-control data <name of the script used in the step above>
```

Shared-Memory Synchronization

Prior to the Shared-Memory Synchronization feature implementation, the data contained in the shared memory (provisioned and dynamic data) of the Cisco BTS 10200 network elements (CA, FS, and so forth) was not periodically written reliably to disk. Because of this, if a power outage occurred and the power was restored, the Cisco BTS 10200 network could come back up on stale data. This could result in loss of service at a customer site, due to loss of provisioned data. Previously, synchronization occurred only at the time of manual platform shutdown.

The Cisco BTS 10200 Shared-Memory Synchronization (SMS) feature adds a periodic synchronization of the shared memory to the disk. If a power cycle happens, the loss of data is limited to the changes since the last sync.

The Shared-Memory Synchronization feature provides the ability to periodically perform a sync of the shared memory to disk at a pre-determined interval. After a power failure, the system will automatically come back up with data from the last sync minimizing data loss.

Troubleshooting

The links below indicate how to troubleshoot the following conditions:

- [Periodic Shared Memory Sync Started—Maintenance \(124\)](#)
- [Periodic Shared Memory Sync Completed—Maintenance \(125\)](#)
- [Periodic Shared Memory Sync Failure—Maintenance \(126\)](#)

Incremental Shared-Memory Restoration

This section describes the Incremental Shared-Memory Restoration (ISMR) feature for Release 6.0 of the Cisco BTS 10200 Softswitch and explains how to use it. The ISMR feature is part of an operational solution that addresses comprehensive recovery from disastrous system incidents like an abrupt power-cut.

The ISMR feature represents a means to recover the Cisco BTS 10200 shared-memory (SHM) on a platform by bringing it back to the state where it was at the time of a system disaster (ex.: power-cut). The Cisco BTS 10200 ISMR works in conjunction with the Cisco BTS 10200 ASMB (Automated Shared-Memory Backup) feature, which restores all of the BTS 10200 shared-memory contents to a point-in-time of the latest SHM backup. However, several provisioning and control commands (incremental commands) might have changed the contents of the SHM after the latest periodic backup. An audit/sync step after an ASMB operation could take considerable time if the incremental commands are not applied in the database in an efficient manner.

The ISMR feature working in conjunction with the Shared-Memory Synchronization (SMS) feature bridges the gap in SHM between the latest backup snapshot and disaster-moment snapshot in an extremely efficient manner. When the disaster-moment snapshot is restored, post-ASMB-backup commands restored, the Cisco BTS 10200 restarted, and EMS database audited/synchronized, the system is ready to restart operations from a system internal disaster incident.

The ISMR feature provides:

- A comprehensive solution for automatic ongoing logging of Post-Backup Incremental Commands (PBIC) (such as provisioning and control commands that were affected in the SHM after latest successful backup).
- A configurable manual incremental restoration of SHM using the incremental commands.

- A best-effort solution with the no specific time constraints. The time for the ISMR to complete depends on the number of PBICs present in the system.
- A complete snapshot of the SHM in conjunction with SMS feature.
- The ISMR feature consists of the following logical components:
 - Recording—The recording component of ISMR feature is responsible to continuously intercept and log the provisioning and control command in a redundant manner to the ISMR log archive.
 - Editing—The editing component of ISMR feature is responsible for sequencing, filtering and formatting an ISMR log file in such a manner that is ready for ISMR replay. The original recording will have certain annotations that need to be filtered out.
 - Playback—The playback component of ISMR feature, which is activated after an ASMB restore, is responsible for performing the necessary editing of the ISMR log archive and for using the logged commands to restore provisioning into the applicable shared memory.

Feature Interactions

The ISMR feature works in conjunction with the ASMB feature and the SMS feature.

Prerequisites

The ASMB feature is a prerequisites for the ISMR feature.

Assumptions

The ISMR feature implementation assumes that all mated-pair Cisco BTS 10200 network-elements work within the same time-zone.

Operating

This section explains how to perform operational tasks for the ISMR feature.

- [Recovery Operations](#)
- [Single Platform Disaster Operations](#)
- [Multi-Platform Disaster Operations](#)

Recovery Operations

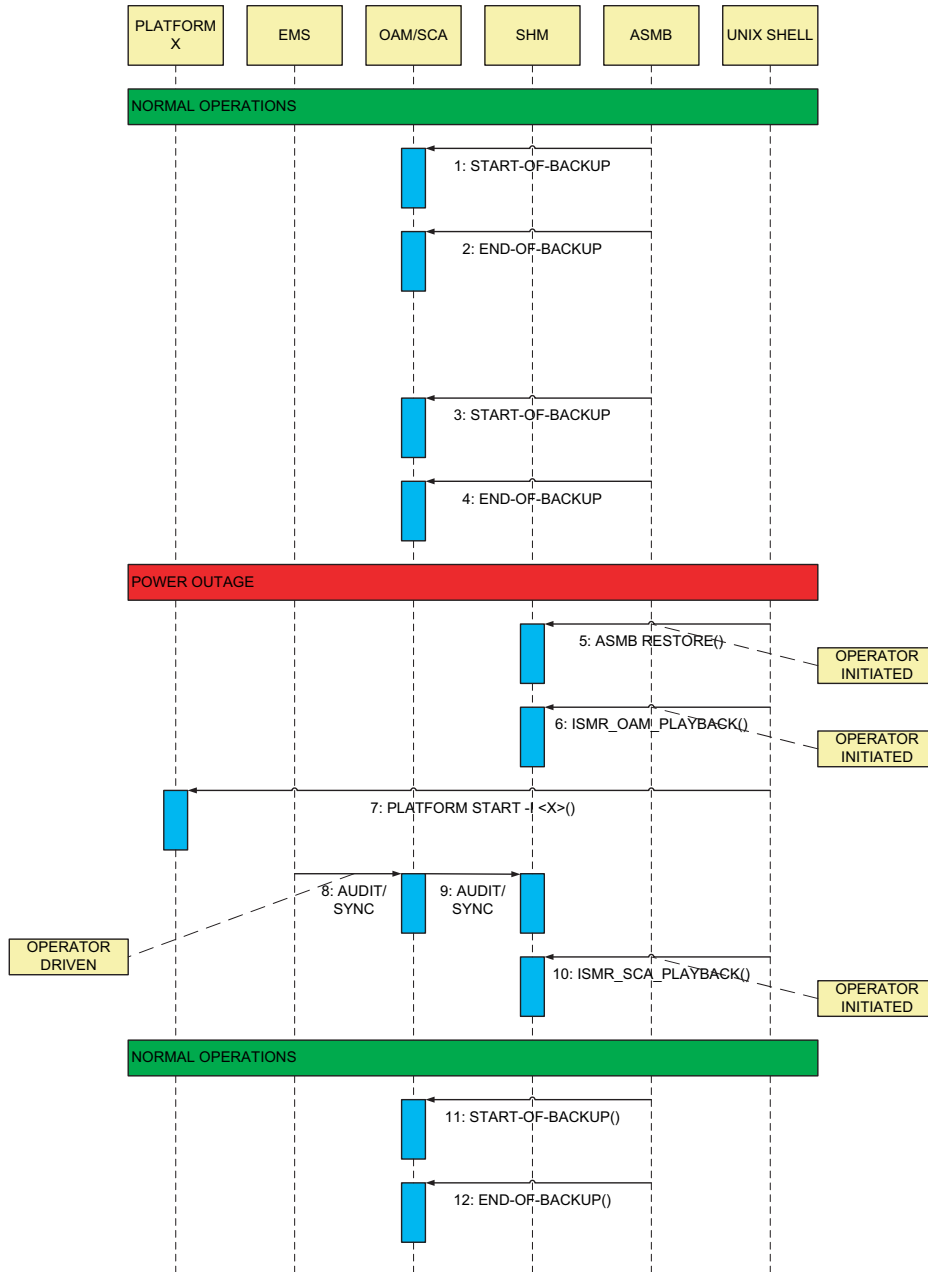
The ISMR recovery operations are shown in [Figure 16-1](#). As a convention, the “()” shown along side tasks in [Figure 16-1](#) implies a predefined set of sub-procedures.



Note

Since at the time of ISMR playback the platform is *not* supposed to be up, there is no active/standby classification. The ISMR provisioning playback should work on either side during this phase. Once the platform is up, ISMR playback will only work on the active side.

Figure 16-1 Comprehensive Shared Memory Backup and Restoration - Operations



Single Platform Disaster Operations

When a single Cisco BTS 10200 platform encounters a disastrous condition resulting in SHM corruption, the following ISMR operational procedures must be followed in order to comprehensively recover from the disaster or SHM corruption:

-
- Step 1** Perform the operational steps recommended for the ASMB feature.
 - Step 2** Execute the **ISMR_prov_playback** command as per its command-line syntax.

- Step 3** Perform a platform start.
- Step 4** Synchronize SHM database with EMS by use of the CLI audit or sync facility.
- Step 5** Execute the **ISMR_ctrl_playback** command as per its command-line syntax.

The Cisco BTS 10200 SHM should now be the same as the post disaster-moment snapshot.

Multi-Platform Disaster Operations

In [Table 16-2](#), a Y in a column indicates that the corresponding platform (in the column header) has been struck with a disaster, while a — indicates normal operation. The fourth column identifies the sequence in which the platforms are restored. The specific recovery steps are noted in the [Single Platform Disaster Operations](#) section. If both primary and secondary sides of a network-element are down, it is recommended to recover the primary side first. In order to save time to restore, CA and FS can be restored simultaneously if they are restored on different hosts. EMS restoration must always follow CA/FS restorations.

Table 16-2 Multi-Platform Disaster Operations Matrix

EMS	CA	FS	Recovery Sequence for Platforms
—	—	—	FS Note If every platform is normal, recovery sequence of platforms should also be normal (-). If needed, all platforms can be included (CA, FS, EMS not only FS).
—	Y	—	CA
—	Y	Y	CA, FS
Y	—	—	EMS
Y	—	Y	FS, EMS
Y	Y	—	CA, EMS
Y	Y	Y	CA, FS, EMS

To troubleshoot the following condition:

- [Secure File Transfer Protocol Transfer Failed—Database \(25\)](#)

See [Secure File Transfer Protocol Transfer Failed—Database \(25\)](#) section in the *Cisco BTS 10200 Softswitch Troubleshooting Guide*.

Disaster Recovery Using the Automatic Shared Memory Back Up



Note

This procedure is for use with Cisco BTS 10200 Release 4.5.1 and above.

This section describes the procedure for restoring a Cisco BTS 10200 network element's shared memory from a back up copy created by the Cisco BTS 10200 Automatic Shared Memory Back Up subsystem.

This procedure should be run only if memory is corrupted in both the active and standby sides of the same Cisco BTS 10200 network element (NE).

**Note**

There is an existing procedure that achieves this same end result. However, it requires a download from the EMS, which can take a relatively long time to complete. The advantage of using the procedure listed here is that the EMS is bypassed, allowing the Call Agent or Feature Server platform to be restored to an operational state in a much shorter time

Additionally, note that the data in the Automatic Shared Memory Back Up (ASMB) can be up to 24 hours old. Provisioning done in the time since the back up will be restored only after an audit and sync with the EMS is completed at the end of the procedure.

Before You Begin

This procedure should be completed only if both of the following conditions are true:

- Both Side A and Side B of the same Cisco BTS 10200 network element are out of service.
- Neither side of the same Cisco BTS 10200 network element can be returned to service with a **platform start** command due to corrupted shared memory

If these conditions are true, then the Cisco BTS 10200 network element should be restored to the ASMB back up copy of the shared memory. If not, then do not perform this procedure.

Before restoring your system, you must have the following:

- Console access to the system to be restored.
- Location of the ASMB shared memory back up for the platform. This back up resides on the system in the /bin directory of each Cisco BTS 10200 network element.

Automatic Shared Memory Back Up Restore

This section describes the procedure for restoring a platform to its ASMB shared memory back up.

It is possible that multiple Cisco BTS 10200 network elements (NEs) need to have the shared memory restored. If so, then this procedure should be executed for each one. It is recommended that the NEs be recovered (if necessary) in the following order:

- Call Agent (CA)
- Feature Server (POTS)
- Feature Server (AIN)
- Element Management System (EMS)
- Billing Data Management System (BDMS)

The EMS and BDMS are usually hosted on a system separate from the Call Agent and Feature Servers; therefore, they can be recovered in parallel with the NEs on the CA/FS system.

Additionally, all Cisco BTS 10200 NEs on one of the systems should be restored first before you attempt to restore those on the mate. This ensures that full service is restored as soon as possible.

**Note**

The backup data directory contains shared memory files in the *gzip* format.

-
- Step 1** Use the console to gain access to Side A and Side B of the system with the network element(s) to be restored.
- Step 2** Verify that both Side A and Side B of the NE are out of service using the **nodestat** command. Verify neither NE will return to service after a **platform start** command. If either side is in service or can be returned to service, exit this procedure.
- Step 3** Choose which side of the NE to restore first. We recommend Side A as a convention, and therefore will refer to Side A in this procedure. Exit from the console of the other system.
- Step 4** On Side A, run the script **restoreSharedMemory**. This script will remove the current, corrupted shared memory and then replace it with the ASMB backup copy. To restore the Call-agent, run **restoreSharedMemory -i CAxxx**.
- Step 5** Bring the Cisco BTS 10200 NE back into service with the **platform start** command.
- Step 6** Verify that the NE returns to service in the Active state using the **nodestat** command.
- Step 7** Repeat [Step 4](#) to [Step 6](#) for FSPTC and FSAIN if necessary using **restoreSharedMemory -i FSPTCyyy** and **restoreSharedMemory -i FSAINzzz**.
- Step 8** After all Cisco BTS 10200 NEs are in service on Side A, perform an audit of the EMS system database to the CA/FS system database. The Audit will report any differences between the two system databases. Because the data in the CA/FS shared memory back up can be up to 24 hours old, differences between the two systems are probable. These should then be resolved with a sync of the EMS database to the CA/FS database.
- Step 9** Perform some test calls to verify that full functionality is provided by the Cisco BTS 10200 after all NEs on Side A have been completely restored. If there are any problems, contact Cisco TAC immediately.

- Step 10** Next, the mate Side B network elements that are out of service need to be restored. For each network element that needs to be restored on Side B, remove its shared memory directory (`./bin/data`) using the unix **rm** command. This should only be done for network elements that are out of service. Start the down NE(s) with the **platform start** command. The network element(s) will start and copy the shared memory database from their Active Side A network element mate(s).
- Step 11** Verify that the mate Side B network elements have all returned to service in the Standby state using the **nodestat** command.
-

Restore Shared Memory Script

This section describes actions performed by the `restoreSharedMemory` script. The `restoreSharedMemory` script completes these actions automatically. These steps do not need to be performed by the user as part of the recovery procedure.

-
- Step 1** Verifies that the platform is OOS-FAULTY. This procedure should not be run otherwise.
- Step 2** Verifies that the installed software version matches the software version of the ASMB shared memory back up copy to be used. The ASMB back up has the software version and timestamp in its name. For example: `data.bak.900-04.05.01.V14.2006_08_29__02_00_06`
- Step 3** Renames the current, corrupted shared memory data directory to keep for potential offline debugging. The software version and the current time are used in the name. For example: `data.corrupt.900-04.05.01.V14.2006_09_01__00_21_50`
- Step 4** Copies the most recent ASMB shared memory back up copy to the data directory for the Cisco BTS 10200 NE being restored.
-

Automatic Restart

This section describes the Cisco BTS 10200 automatic restart feature. The Cisco BTS 10200 automatic restart feature is beneficial to customers because the Cisco BTS 10200 will attempt to automatically restart a platform (EMS/FS/CA) to standby that has become OOS-faulty and has shut down. Currently the Cisco BTS 10200 will not restart a platform in this situation, leaving the active platform running in a vulnerable simplex mode until the standby platform is restored manually.

**Note**

An automatic restart is not attempted if the restart of the system is likely to fail.

Benefits of this feature:

- **Reduced outage risk.** Automatic restart—when successful—brings the platform up to standby in minutes instead of potentially much longer, as support personnel work to restore the platform. This reduces the risk of outages by reducing the amount of time the system is in simplex mode.
- **Automated forensic data collection.** This feature automatically saves data useful for offline debugging (trace logs, status files, cores, and so on). Currently this data is retrieved manually. Automating the process of saving off useful information guarantees that this information is preserved.
- **Faster switch overs.** When a process exceeds the maximum number of restarts, a system-initiated switchover is executed. By not transitioning to OOS-faulty, the standby side avoids the taxing database copy process.

The processing associated with the Cisco BTS 10200 automatic restart feature has two main phases. In the first phase, the transition is made to OOS-faulty. During this phase, the forensic information for offline analysis must be collected and stored. In the second phase, the attempts are made to bring the platform from OOS-faulty to standby.

Transition to OOS-Faulty

When the platform is transitioning to OOS-faulty, the automatic restart feature requires the additional processing described here.

There is no need to save the alarm log or event log because they are contained in the MySQL database. This database is external to the platform and is not affected by platform shutdowns or bring ups.

The following processes are completed during the transition to OOS-faulty:

- **Create /saved.debug Subdirectory**—Create the /saved.debug subdirectory.
- **Archive the /data Directory**—Copy and archive the /data directory and all of its contents to the /saved.debug directory using a tar file with a timestamp in the name.
- **Unwritten Traces**—Any unwritten, buffered trace memory is saved.
- **Save Trace Logs**—The most recent trace log files are saved. The number of logs to save is a configurable parameter. This is especially useful when the Automatic Restart feature is active because a system side might only temporarily remain in OOS-faulty. If the system side comes back to a working standby state, it is less urgent that support personnel retrieve logs, and so on, from the node.
- **Disk Space for /saved.debug**—The disk space consumed by /saved.debug is conserved by deleting the /saved.debug directory and its contents each automatic restart cycle. The saved.debug subdirectory is created in the platform's /bin directory, which is <path>/bin/saved.debug.

Automatic Restart Processing

Automatic restart processing is not always desired, or possible. Table 16-3 lists the platform restart actions for the OOS-faulty state. The restart actions fall into one of three categories:

- SDAMR (Shut Down and Manually Restart)—These scenarios are handled the same way as the OOS-faulty behavior for earlier releases of the Cisco BTS 10200. The platform will shut down and wait for manual restart. All initialization-related processing (bring up) is treated this way.
- SDAAR (Shut Down and Automatically Restart)—These are the scenarios in which an automatic restart is attempted.
- FISS (Fault-initiated Switch to Standby)—In these scenarios, a faster system-initiated switchover is attempted.

The scenarios are grouped into PMG-initiated, KAM-initiated, and application-initiated. PMG and KAM start the vast majority of shutdowns. For PMG-initiated and KAM-initiated, the shutdowns are grouped by the state of the platform.

Table 16-3 OOS-Faulty Behavior

Description	Result
PMG-initiated Shutdowns	
Bringup	
One or more processes did not come up to correct state. ¹	SDAMR
Unable to open log file (/var/adm/events_to_<plat>).	SDAMR
During bring up to active, one or more processes in a bad state.	SDAMR
During bring up to standby, one or more processes in a bad state.	SDAMR
One or more processes in a bad state.	SDAMR
Active	
A crashing process exceeded the restart rate.	See Note ²
Disk usage exceeded.	SDAMR
Swap space usage exceeded and mate ok.	SDAAR
IPC memory usage exceeded and mate is ok. If mate not ok, no shutdown.	SDAAR
Standby	
A crashing process exceeded the restart rate.	See Note 2
IPC memory usage exceeded and mate is ok. If mate not ok, no shutdown.	SDAAR
Disk usage exceeded.	SDAMR
Swap space usage exceeded and mate ok.	SDAAR
Switchover to Active	
Switchover to active: failed to start all processes in active.	SDAAR
Switchover to standby	
Switchover to standby: failed to start all processes in standby.	SDAAR
All States	
Watchdog expires for a PMG thread—or threads that are not PMG threads handled by restart logic.	SDAAR

Table 16-3 OOS-Faulty Behavior (continued)

Description	Result
When PMG receives a SIGINT.	SDAAR
KAM-initiated Shutdowns	
Bringup	
System time gets out-of-sync with mate when standby comes up.	SDAMR
Network interface is down.	SDAMR
Timer initialization failure.	SDAMR
Configuration error with ip addresses.	SDAMR
Thread creation failed.	SDAMR
Shared memory DB in bad state.	SDAMR
Data replication not enabled to run.	SDAMR
Mate faulty during DB copy.	SDAMR
DB copy fail.	SDAMR
DB clear timeout.	SDAMR
DB copy timeout.	SDAMR
Data replication connection down.	SDAMR
Init thread failed.	SDAMR
HB timer failure.	SDAMR
Time out of sync with mate.	SDAMR
Mate wrong side.	SDAMR
Mate wrong ip address.	SDAMR
Mate standby state.	SDAMR
Mate not active.	SDAMR
Active	
Active–heartbeat failure and mate is OK.	SDAAR
Standby	
Heartbeat with mate not working properly.	SDAAR
Timeout waiting for call data updates completion.	SDAAR
DB copy timeout.	SDAAR
DB copy failed.	SDAAR
Switchover: failed to receive completion acknowledgement.	SDAAR
Unable to establish data replication connection to mate.	SDAAR
While waiting for CRQ, data replication was restarted.	SDAAR
Switchover to Active	
Switchover to active, but failure.	SDAAR
Switchover to active, but timeout failure.	SDAAR
Switchover to active, but RDM restarted.	SDAAR

Table 16-3 OOS-Faulty Behavior (continued)

Description	Result
Switchover to Standby	
Switchover to standby, but failure.	SDAAR
Switchover to standby, but timeout failure.	SDAAR
Switchover to standby, but DB copy failure.	SDAAR
All States	
KAM received a signal.	SDAAR
Application-initiated Shutdowns	
Active	
BMG2 (Billing Manager). Problems parsing command line options.	See Note 2
H3A. When there are no good H3A signaling links.	See Note 2
MGA. When all the needed interfaces are either faulty or not stabilized.	See Note 2
SGA. When both signaling gateways are down.	See Note 2
EPA. When the EMS is not reachable.	See Note 2
NIM. When all of the HUB interfaces are faulty.	See Note 2
Standby	
BMG2 (Billing Manager). Problems parsing command line options.	See Note 2
H3A. When there are no good H3A signaling links.	See Note 2
MGA. When all the needed interfaces are either faulty or not stabilized.	See Note 2
SGA. When both signaling gateways are down.	See Note 2
EPA. When the EMS is not reachable.	See Note 2
NIM. When all of the HUB interfaces are faulty.	See Note 2

1. Most references to process state in this table refer to a platform-internal state assigned to each process. The Process Manager methodically brings up the platform by starting processes in a given sequence. When a process does not reach the proper state at a given point during startup, it is declared to be in a bad/or incorrect state.
2. The shutdown action is specified by the process's platform.cfg parameters: ProcessCriticalFailureActionWhenMateStandby, ProcessCriticalFailureActionWhenMateActive, or ProcessCriticalFailureActionWhenMateFaulty.

The following processes are completed during the automatic restart processing:

- Check if Restart is Possible (Table 16-3)—An intelligent processing to check if a restart should be attempted is completed. The default behavior is to attempt an Automatic Restart the maximum number of configured times.
- platform start/platform stop—The **platform start** and **platform stop** commands do not work when an Automatic Restart is scheduled. However, the user can use the **-noautorestart** command option on either command to bypass the Automatic Restart and force the platform start or stop to occur immediately.

Installing

There are two customer-provisionable parameters (SystemAutoRestartRate and SystemAutoRestartDelay) and one new platform-provisionable parameter (SystemAutoRestartNumSavedTraceLogs) for the Cisco BTS 10200 automatic restart feature. Additionally, two existing parameters are expanded into three parameters.

Configuring

This section explains how to perform the following tasks:

- [Optical Configuration](#)
- [Platform Configuration](#)

Optical Configuration

The configuration changes for this feature are the new customer-configurable parameters added to the optical.cfg file. These parameters are listed and discussed here. They are used to control the Automatic Restart feature.

- SystemAutoRestartRate—This variable indicates the number of times a platform auto restart is attempted. This variable is defined to be the number of times a restart shall be attempted over a 30-minute interval.
- SystemAutoRestartDelay—A side that is down should delay by this amount before attempting to restart. The default is 10 minutes. The newly active side is allowed time to begin processing fully before needing to support the auto restart of the faulty side.

Platform Configuration

The following parameter is added to the platform.cfg file.

- SystemAutoRestartNumSavedTraceLogs—This variable indicates the number of most recent consecutive trace logs to save.

The following parameters are changed in platform.cfg file. In earlier releases, these parameters could be set for each process.

- SwitchOverIfMaxRestartExceededInDuplex = [yes | no]
- EndPlatformIfMaxRestartExceededWhenMateFaulty = [yes | no]

To support the Automatic Restart feature, these two parameters are replaced with the following three parameters.

- ProcessCriticalFailureActionWhenMateStandby = [NONE | SDAMR | SDAAR | FISS]
- ProcessCriticalFailureActionWhenMateActive = [NONE | SDAMR | SDAAR]
- ProcessCriticalFailureActionWhenMateFaulty = [NONE | SDAMR | SDAAR]

These parameters allow different types of Automatic Restart shutdowns to be specified when a process encounters a critical fault in Active, Standby, or Mate Faulty. For example, if the NIM (Node Interface Monitor) encounters a critical fault in Active, a FISS could be specified, but if it encounters the fault in Standby, an SDAAR could be specified.

FISS is not an option when the process is in a Standby state or a Mate Faulty state.

Troubleshooting

**Note**

The following troubleshooting procedure can be used any time a platform side goes OOS due to a fault.

Troubleshooting of a platform shutdown can be completed using the Automatic Restart feature saved.debug directory. When a platform shutdown occurs, useful debugging information (cores, logs, and so on) is saved and compressed in a tar file. This file is located in the directory <platform>/bin/saved.debug. To recover the tar file information and to clear the tar file disk space, complete the following steps:

-
- Step 1** Download the time stamped tar file from the saved.debug directory to a system for offline debugging. An example filename is: saved.debug.2007_02_16__16_07_09.tar.Z.
- Step 2** Remove the tar file from the disk after it is downloaded to conserve disk space.
-

Switchover in Progress–Maintenance (101)

The Switchover in Progress alarm (critical) indicates that a system switchover is in progress. This alarm is issued when a system switchover is in progress either due to manual switchover (by a CLI command), failover switchover, or automatic switchover. For additional information, refer to [“Maintenance \(101\)” section on page 7-56](#).

Side Automatically Restarting Due to Fault–Maintenance (117)

The Side Automatically Restarting Due to Fault alarm (critical) indicates that a platform side is automatically restarting due to a fault condition. This alarm indicates that an Automatic Restart is pending and at what time it will be attempted. For additional information, refer to [“Maintenance \(117\)” section on page 7-64](#).

Sh Interface Troubleshooting

This section explains how to troubleshoot disaster recovery.

Disaster Recovery

The Diameter Request policy PRIORITY-ORDER means the BTS 10200 always selects the Diameter peer with the highest priority as long as the peer is operational. It prevents switchover if all the peers go down. If the BTS 10200 cannot establish a connection with the highest priority peer, it resends the pending transactions to the peer with the second-highest priority. If the higher priority peer comes back up, the BTS 10200 sends new Diameter requests to the higher priority peer.

During switchover the BTS 10200 does the following:

1. Tears down Diameter connections on the previously-active side.

2. Creates Diameter connections from newly-active side to all Diameter peers IN-SERVICE.
or
If peers are unreachable, the BTS 10200 active side does not switchover. Instead it attempts to reestablish the connection with its peers using DIA_REATTEMPT-INTERVAL and DIA_RETRY_COUNT.
DIA_REATTEMPT-INTERVAL—number of seconds until the DIA_RETRY_COUNT is DIA_RETRY_COUNT
3. The BTS 10200 active side re-attempts to setup the Diameter connection.

