Troubleshooting Cisco BAMS

Revised: April, 2010, OL-11618-16

Introduction

This appendix provides a discussion of system tasks and messages generated by the Cisco Billing and Measurements Server (BAMS) and their use in troubleshooting. These messages, which can be categorized as alarms or events, are generated by the different software tasks running on BAMS. This appendix also provides information on the file-naming conventions used for files collected or processed by BAMS.

BAMS traps alarms and minor, major, and critical events, and forwards them to the operations management system (OMS).

BAMS writes alarms only to the alarm history file. When an alarm is cleared, its entry is deleted from that file.

Alarms and minor, major, or critical event messages to be forwarded to a message queue and trapped are defined with the Alarm Parameters table (ALM-PARMS tag ID), with the msgfwdlevel field (see the "ALM-PARMS Tag ID" section on page 5-2 for details).

> **Note** All task messages are written to the system log (syslog) file. We recommend that the operations support personnel monitor the syslog file on a regular basis and note alarms and events that require troubleshooting. The bootstrap function can facilitate this task.

> **Note** The following alarms are cleared when the alarm condition is removed: CRD502, MGR801, MSC030, MSC031, MSC032, MSC040, MSC041, MSC042, MSC130, MSC131, MSC132, MSC140, MSC141, MSC142, POL327, POL361, POL401, POL402, and POL115.

System Tasks

BAMS functions are governed by system tasks. Table A-1 displays BAMS non-SNMP tasks, and Table A-2 displays BAMS SNMP tasks. For information about the messages produced by these system tasks, see the "Generic Messages" section on page A-12.

Table A-1   BAMS Non-SNMP Tasks

Task	Explanation
ACC	The Accumulation task. This task accumulates numerical information on collected usage records.
ALM	The Alarm task. This task collects and distributes alarm and event information.
AUG (AGB)	The Augmentation task. This task uses lookup information to add value to the collected usage records. Messages related to BAF output are written to the system log with an Augmented BAF (AGB) task identifier.
ASC	The ASCII conversion task. This task converts the Cisco Media Gateway Controller (MGC) call detail record (CDR) files to ASCII files.
BIN	The BIN1110 output task. This task generates data in 1110 Binary format.
BTP	The bamstrap task. This task converts alarm messages into traps.
COR	The Correlation task. This task correlates records received from each Cisco MGC in a redundant pair configuration and produces a single-record format.
CRD	The Check Remote Down task. This task monitors communications between the two BAMS units and checks the status of the remote data collector.
CTB	The CDR to Bellcore AMA Format (BAF) conversion task. This task generates CDRs in BAF format.
DRX	The Message Receive task. This task is used by one BAMS unit to receive messages from another BAMS unit.
DTX	The Message Transmit task. This task transmits communications from one data collector to the next.
EXT	The Extendable ASCII task. This task generates data in extendable ASCII format.
FMT	The Formatting task. This task controls the initial formatting of data collected from the Cisco MGCs.
MGR	The Manager task. This task manages and spawns all other BAMS tasks.
MML (mml)	The Man-Machine Language (MML) task. This task displays the MML interface.
MSC	The Mass Storage Control task. This task manages the user-defined attributes governing the maintenance and removal of files from BAMS.
NIC	The NICS output task. This task generates data in NICS format.
NNL	The Node Name Link task. This task synchronizes MML commands to the remote BAMS unit.
POL	The Redundant Unit Polling task. This task collects data from (redundant) Cisco MGC units.

Table A-1   BAMS Non-SNMP Tasks (continued)

Task	Explanation
RCV	The Control File Receive task. This task is used by the BAMS unit to receive a control file. This task is executed by the POL task and is not a full-time running task.
SXF	The Send File Transfer task. This task sends a control file to a remote BAMS unit. This task is executed by the POL task and is not a full-time running task.

Table A-2   BAMS SNMP Tasks

Task	Explanation
bamsP3MIBagt	Sends out BAMS alarm messages as SNMP traps.
mscP3MIBagt	Returns parameter values of the MSC Thresholds table.
mscparmP3MIBagt	Returns parameter values of the MSC Parameters table.
nodeparmP3MIBagt	Returns parameter values of the Node Parameters table. This task also returns node name and poll status information for each node.
pollP3MIBagt	Returns parameter values of the Poll table.

Procedure to Recover from Active/Active Polling State

The normal state of the redundant BAMS is active/standby, where one unit is actively polling the Cisco MGC and the other unit is in a standby mode. Under certain error conditions, the BAMS unit can enter an active/active polling state. The following procedure explains how to recover from the undesired active/active polling state.

The polling software was designed so the worst-case scenario would be a standby/standby state, because in this state neither BAMS unit is polling and data is accumulating on the VSC. In safeguarding the system so a standby/standby state would not occur, certain race conditions (where a possible standby/standby state might occur) have been taken care of by defaulting to an active/active polling state. There are few cases in which this would happen, but defaulting to an active/active polling state is a safe method of avoiding any critical problems.

The active/active polling state is not a critical state because in this mode, the BAMS unit that has the correct sequence numbers can continue polling, and the other BAMS unit, although in an active state, cannot poll the files that the first BAMS has already polled. Also, the processing would not be interrupted because the BAMS unit that can poll forwards the data to the other BAMS unit for redundant processing.

To reset the system to a normal active/standby state, you must update the system that is not polling and has the incorrect sequence numbers. Updating can be accomplished by executing a rotation (sw-ovr) from active to standby on the BAMS unit that has the correct sequence numbers; that is, the unit that is actually polling files. The rotation from active to standby causes this BAMS unit to go to standby, and the other BAMS unit that was active but was not able to poll is updated with the correct polling sequence numbers. Once the sw-ovr is executed, the active unit can poll the VSC successfully, and a normal active/standby state results.

File-Naming Conventions

For consistency, the application requires that the data sources follow certain file-naming conventions. The convention is to use a unique prefix per file format, followed by a time stamp that indicates when the file was created or generated on the data source, and a file sequence number.

Input and Output File-Naming Conventions

The following file-naming conventions are used for BAMS input and output files.

ASCII BAF Output

The file-naming convention for data collected from the Cisco MGC and converted into ASCII BAF record format is as follows:

baf_<raw file prefix>_YYYYMMDDHHMMSS_SeqNo.csv

Table A-3 explains each element of the filename. Note the use of the underscore.

Table A-3   ASCII BAF Filename Elements

Element	Description
baf	Represents the fixed prefix for ASCII BAF output files
raw file prefix	Represents the raw filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.csv	Represents the suffix

BAF Output

The file-naming convention for BAF billing record files is as follows:

baf_<raw file prefix>_YYYYMMDDHHMMSS_SeqNo.bin

Table A-4 explains each element of the filename. Note the use of the underscore.

Table A-4   BAF Billing Record Filename Elements

Element	Description
baf	Represents the fixed prefix for BAF output files
raw file prefix	Represents the raw filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004

Table A-4   BAF Billing Record Filename Elements (continued)

Element	Description
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.bin	Represents the suffix

Investigating BAMS CDR Output Records

To investigate BAMS billing output records, start by retrieving the archived PGW CDR records, which are located in the node TLV directory. The TLV directory contains most of the information that populates the BAMS output. It is important to notice that the sequence number of the file in the TLV directory may not directly correspond to the sequence number of the BAMS output file. Furthermore, the sequence of individual records recorded in the BAMS file may not be identical to the sequence of records recorded in the TLV file.

BIN1110 Output

The file-naming convention for data collected from the Cisco MGC and converted into 1110 Binary record format is as follows:

cdr_YYYYMMDDHHMMSS_SeqNo.BIN

Table A-5 explains each element of the filename. Note the use of the underscore.

Table A-5   BIN1110 Filename Elements

Element	Description
cdr_	Represents the prefix of the files generated by the Cisco MGC.
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.BIN	Represents the suffix (specified in the BIN1110 table)

Extendable ASCII Output

The file-naming convention for data collected from the Cisco MGC and converted into extendable ASCII record format is as follows:

extasc_<raw file prefix>_YYYYMMDDHHMMSS_SeqNo.csv

Table A-6 explains each element of the filename. Note the use of the underscore.

Table A-6   Extendable ASCII Filename Elements

Element	Description
extasc	Represents the fixed prefix for EXTASCII output files
raw file prefix	Represents the raw filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.csv	Represents the suffix

Input CDR Files

The file-naming convention for data collected from the Cisco MGC is as follows:

prefix_YYYYMMDDHHMMSS_SeqNo.bin

Table A-7 explains each element of the filename. Note the use of the underscore.

Note After the file is polled from the Cisco MGC, FMT (the first BAMS processing task) renames it according to the GMT timestamp inside the file (CDB tag 400 of the file header record). For this reason there may be a slight discrepancy between the original timestamp in the filename and the timestamp in the BAMS filename. All BAMS tasks that process the file after this point use the GMT-based filename.

Table A-7   CDR Filename Elements

Element	Description
prefix	Represents the filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23

Table A-7   CDR Filename Elements (continued)

Element	Description
SeqNo	Represents the sequence number; for example, _000051
.bin	Represents the suffix

MGC End of Call 1110 ASCII Output

The file-naming convention for data collected from the Cisco MGC and converted into ASCII 1110 record format is as follows:

prefix_YYYYMMDDHHMMSS_SeqNo.csv

Table A-8 explains each element of the filename. Note the use of the underscore.

Table A-8   ASCII 1110 Filename Elements

Element	Description
prefix	Represents the filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.csv	Represents the suffix

NICS Output

The file-naming convention for data collected from the Cisco MGC and converted into NICS record format is as follows:

CDR.<switch_name>.<SeqNo>.YYYYMMDDHHMMSS

Table A-9 explains each element of the filename. Note the use of the underscore.

Table A-9   NICS Filename Elements

Element	Description
CDR.	Represents the fixed prefix for NICS output files
switch_name	Represents the switch name. It is taken from TAG6000 of the data produced by the PGW (that is, it denotes the configured switch name for the PGW)
SeqNo	Represents the sequence number; for example, 1234
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15

Table A-9   NICS Filename Elements (continued)

Element	Description
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23

P01 Output

The file-naming convention for data collected from the Cisco MGC and converted into the P01 record format is as follows:

p01_<raw file prefix>_YYYYMMDDHHMMSS_SeqNo.bin

Table A-10 explains each element of the filename. Note the use of the underscore.

Table A-10   P01 Filename Elements

Element	Description
p01	Represents the fixed prefix for P01 output files
raw file prefix	Represents the raw filename prefix; for example, cdr_
YYYY	Represents the year; for example, 2004
MM	Represents the month; for example, 05
DD	Represents the day; for example, 15
HH	Represents the hour; for example, 19
MM	Represents the minute; for example, 46
SS	Represents the second; for example, 23
SeqNo	Represents the sequence number; for example, _000051
.bin	Represents the suffix

Task Log Files

Log files are used to save debug or status information for a particular task. These files are saved to the /opt/CiscoBAMS/files/s01-s08 individual node directories. Here is the filename format:

task-identifier.log

Table A-11 explains each element of the filename. Note the use of the underscore.

Table A-11   Task Log Filename Elements

Element	Description
task_identifier	Tag to identify the task to which this log file refers; for example, the FMT task. Sometimes followed by a qualifier; for example, _cdr.
.log	Fixed suffix to indicate that the file is a log file.

Task Configuration Files

Task configuration files pass all the necessary command-line switches that are needed by the MGR task to properly execute a particular task. These files are stored in the /opt/CiscoBAMS/files/s01-s08 individual node directories. Here is the filename format:

*task_identifier{task_instance}_*cfg

Table A-12 explains each element of the filename. Note the use of the underscore.

Table A-12   Task Configuration Filename Elements

Element	Description
task_identifier	Tag to identify the task to which this configuration file refers.
task_instance	Optional instance sequence. For example, if there are two instances of the COR task running, the filenames might be named cor0_cfg and cor1_cfg.
_cfg	Fixed suffix to indicate that the file is a task configuration file.

Task Parameter Files

Task parameter files contain parameter definitions for a particular task. These files are stored in the /opt/CiscoBAMS/files directory. These files are in ASCII format. The filenames used are shown in Table A-13. Here is the filename format:

task_identifier{*task_instance*}.cfg

Table A-13   Task Parameter Filename Elements

Element	Description
task_identifier	Tag to identify the task to which this configuration file refers.
task_instance	Optional instance sequence. For example, if there are two instances of the COR task running, the filenames might be named cor0.cfg and cor1.cfg.
.cfg	Fixed suffix to indicate that the file is a task configuration file.

System Log File (syslog)

BAMS stores system activity in the system log file (syslog) in the format explained in Table A-14. Located in the /CiscoBAMS/files directory, the syslog is composed of individual log entries that follow this format example: *MM/DD/YY hh:mm:ss severity message ID: +message text*.

Table A-14   Syslog Format

Item	Description
MM/DD/YY	The date that BAMS wrote the entry to the file.
hh:mm:ss	The time that BAMS wrote the entry to the file.

Table A-14   Syslog Format (continued)

Item	Description
severity	If indicated, represents the importance level of the message:
	*C indicates a critical event or alarm.
	** indicates a major event or alarm.
	* indicates a minor event or alarm.
	*R indicates a resolved event or alarm.
	W indicates a warning event.
	M indicates a manual problem (operator intervention required).
	Two blank spaces indicate an informational event.
	DB indicates a debug event.
message ID	A three-character task name that generated the message, followed by a three-digit code that identifies the particular message.
message text	The brief message or description of the alarm or event. If a plus sign (+) is included it indicates that the line is a continuation of the previous line.

Log File History

The syslog file holds a user-specified number of messages. When the syslog file's maximum file size is reached, BAMS changes the name of the syslog file to oldlog and creates a new syslog file.

While only one syslog file can exist at a time, BAMS can retain multiple oldlog files. BAMS creates additional oldlog files in the same manner that it creates syslog files. If an oldlog file exists when it is time to create a syslog, BAMS renames the oldlog file to oldloga. If more than two oldlog files exist, BAMS uses the letters b, c, d, and so on to identify the oldest oldlog file. (For example, oldlogd is older than oldlogc.)

Message Description

The following categories of information are provided for each system message:

- Message ID
- Text
- Arguments
- Description
- Action

Message ID

The message ID is a six-character label (three characters followed by three digits) that uniquely identifies each message. The first three characters are the task ID, which identifies the task that generated the message (for example, MGR denotes the Manager task and MSC denotes the Mass Storage Control task). The three digits represent the message number (for example, 013 or 123).

> **Note** In the task message section below, minor, major, critical, and warning alarms are preceded by a one- or two-character signifier (* = minor, ** = major, *C = critical, and W = Warning) before the message ID. In some cases, such as for MSC alarms, a message may be generated as minor, major, or critical, depending on the system state. In such cases, the range of levels is represented between parentheses: (*, **, *C). While informational messages are preceded by two blank spaces in the syslog, they are not preceded by any characters in the task message section.

Text

The text is the verbal part of the message as it appears on your screen or in the system log file.

Arguments

Arguments are variable parts of the message (message arguments) and are enclosed in angle brackets (< >). Explanations of arguments are provided.

Description

The message description explains the event(s) that generated the message.

Action

The action describes what you should do as a result of the event described in the message. In some cases (for example, informational messages), no action might be required. Actions for error messages (manual, warning, minor, major, and critical) might include steps that should be followed so that you can identify and correct problems. Error actions can also describe how BAMS responds to the specified error condition.

Alarm Task Messages

The Alarm (ALM) task handles all alarm and event messages that are written to the system log. This includes clearable alarms that are based on user-defined thresholds and are written to an alarm history file.

```
ALM100 ALM Version <version> has started.
```
The <version> argument represents the task version. This message indicates that the ALM task has started. This is an informational message. No action is necessary.

```
* ALM102 <routine> failed on error <UNIX error number>
```
The <routine> argument represents the routine name. The <UNIX error number> argument represents the UNIX error set by the system function. This message indicates that a system failure occurred. Check the UNIX error number for an explanation of this error.

```
** ALM103 <string>() failed on error <number>
```
The <number> represents the UNIX error number. This message indicates that the ALM task encountered a serious error condition. This is a major event. Examine the error number and contact the Cisco Technical Assistance Center.

ALM104 get_qid (<name>) returned <error> (errno <UNIX error number>)
The <name> argument represents the queue ID (QID). The <error> argument represents the returned error. The <UNIX error number> argument represents the UNIX error set by the system function. This message indicates that BAMS could not get the QID from the acecqid file. Verify that the $FILES environment variable is located in the acecqid file. Make sure that the read permissions are correct.

* ALM105 msgget <function> failed (errno <UNIX error number>)
The <function> argument represents the function name. The <UNIX error number> argument represents the UNIX error set by the system function. This message indicates that the ALM task experienced an error. Check the UNIX error number for an explanation of this error.

* ALM106 Can not load <name>
The <name> argument represents the parameter name. This message indicates that the ALM task experienced an error.

* ALM108 Cannot access <string> (errno <UNIX error number>)
The <UNIX error number> argument represents the UNIX error set by the system function. This message indicates that the ALM task experienced an error. Check the UNIX error number for an explanation of this error.

* ALM109 Alarm table is full
The number of alarms set in the BAMS system has exceeded the maximum limit for the system. Use the MML command **rtrv-alms** to view the set alarms. If the maximum limit is reached, it is likely that the alarm ACC227 is being set. Review and modify the measurement alarm threshold definitions in the TCA-TBL table so that the number of measurement alarms being triggered is lowered. You can raise the threshold at which an alarm is generated.

ALM120 Alarm parameters have been updated
This message indicates that alarm parameters have been updated. This is an informational message. No action is necessary.

Generic Messages

The MGR task controls BAMS operations and uses what is called generic messaging. This means that message numbers can be used by more than one software task (ACC001, BTP001, and MGR001, for example). The text of the message is always the same, although the arguments, if any, might be different. The system messages are numbered from 000 through 399 (not all numbers are used).

Note All system messages start with a date and time stamp (not shown below). The standard three-character task identifier is denoted by XXX in the generic messages that follow.

XXX000 <task> Version <number> has started, pid <ID>.
The <task> argument represents the task mnemonic, <version> represents the task version number, and <ID> represents the UNIX process ID. The task logs this message when it is started.

XXX001 <task> Initialization complete.
The <task> argument represents the task mnemonic. The task logs this message when initialization is complete.

XXX002 Task completed, pid <process id>
The <process id> argument represents the UNIX process ID. The task has completed its job successfully.

(*,**, or *C) XXX003 Exiting with error code <error code>, pid <process id>
The <error code> argument represents the error code, and <process id> represents the UNIX process ID. The task is ending due to an error. Refer to a previous message from the same task in the syslog to get more detailed information.

```
(*,**,or *C) XXX004 Terminating on signal <signal code>, pid <process id>
```
The <signal code> argument represents the signal code, and <process id> represents the UNIX process ID. The task is ending abnormally due to the receipt of an unexpected signal. Refer to a previous message by the same task in the syslog to get more detailed information.

```
XXX005 Sending <message> to <task or task component>
```
The <message> argument represents the information or message sent, and <task> represents the task or task component recipient. The task sent information or a message to another task or task component.

```
XXX006 Waiting for <message> message
```
The <message> argument represents the message being waited for. The task is attempting to receive some information from another task or task component.

```
XXX007 Received <item>
```
The <item> argument represents the item or message received. The task has received some information from another task or task component.

```
XXX008 Processing <item to process>
```
The <item to process> argument represents the item to process, such as filename or message type name. The task is processing an item.

```
XXX009 Successfully processed <processed item>
```
The <processed item> argument represents the processed item. The task has successfully processed the indicated item.

```
XXX010 System state is <system state>
```
The <system state> argument represents the system state (for example, Running, Booting, Active, or Standby). For a redundant system, this message is displayed by the task, if its actions are dependent on the system state.

```
XXX011 SIG setting signal <signal number> handler to <signal handler name>
```
The <signal number> argument represents the signal number, and <signal handler name> represents the signal handler name. The task is setting the handler for the given signal.

```
XXX012 SHM Creating shared memory key= <shared memory key> size= <size of shared memory
requested>
```
The <shared memory key> argument represents the shared memory key, and <size of shared memory requested> represents the size of shared memory requested. The task allocates shared memory as a mechanism for interprocess communication.

```
XXX013 SHM Attaching to shared memory key= <shared memory key> size= <size of shared
memory>
```
The <shared memory key> argument represents the shared memory key, and <size of shared memory> represents the size of shared memory. The task attaches to a region in shared memory.

```
XXX014 MSGQ Creating message queue key= <message queue key>
```
The <message queue key> argument represents the message queue key. The task allocates a message queue as a mechanism for interprocess communication.

```
XXX015 SEMA Creating semaphore key= <semaphore key>
```
The <semaphore key> argument represents the semaphore key. The task allocates a semaphore as a mechanism for interprocess communication.

```
XXX016 DIR Changing to <directory name>
```
The <directory name> argument represents the directory name. The task is setting its current work directory.

```
XXX017 Renaming <original filename> to <new filename>
```
The <original filename> argument represents the original filename, and <new filename> represents the new filename. The task renames (or move) a file, typically to mark it as processed.

```
XXX018 Connecting to <remote host address or name> / <port number or service name>
```

The <remote host address or name> argument represents the remote host address or name, and <port number or service name> represents the port number or service name. The task attempts to connect to the remote host (or server). If the task uses a TCP/IP socket and the second argument is supplied as a service name, the service must be registered in the /etc/services file. In addition, when the host name is used instead of an IP address, the host name must be registered in the /etc/hosts file.

```
XXX019 Connected to <remote host address or name> / <port number or service name>
```
The <remote host address or name> argument represents the remote host address or name, and <port number or service name> represents the port number or service name. The task has successfully connected to the named host. At this point, application-layer communication can continue.

```
XXX020 Disconnecting from <remote host address or name> / <port number or service name>
```
The <remote host address or name> argument represents the remote host address or name, and <port number or service name> represents the port number or service name. The task closes the connection to the remote server. To find out why the task is disconnecting, check previous messages in the syslog file.

```
XXX100 Timed out waiting for <description>.
```
The <description> argument represents the description of what the task was waiting for. The task logs a message when the attempt to receive some information from another task or task component has exceeded a certain time threshold.

```
(*, **, or *C) XXX101 Invalid command-line option, <unsupported character>.
```
The <unsupported character> argument represents the unsupported character command-line switch. The task was passed an unknown or unsupported command-line argument. The configuration file for the task contains an invalid option. To check all available options, invoke the task using the **-h** option, and check the contents of the configuration file.

```
(*, **, or C) XXX102 Unrecoverable failure.
```
The task cannot recover from an error. Analyze previous or following entries in the syslog file. These messages provide more information about the possible cause of the failure. Contact the Cisco Technical Assistance Center.

```
(*, **, *C) XXX103 SIG errno <error number> setting handler for signal <signal number>
```
The <error number> argument represents the UNIX error number, and <signal number> represents the signal number. The task failed to set the handler for the given signal. Contact the Cisco Technical Assistance Center if necessary.

```
(*, **, *C) XXX104 SYS errno <error number> on system() for <argument>
```
The <error number> argument represents the UNIX error number, and <argument> represents the description of the argument passed to the system() call. The task detected an error when passing a command to the UNIX shell. The shell might be unable to find and execute the specified command. Check if the command or script passed to the system call is valid. If it is a script, verify that the execute permissions are set. Another possible reason for a system call to fail is an excessive number of UNIX processes already executing. To see if this is the case, issue the **ps -eaf | pg** command.

```
(*, **, *C)XXX105 SYS errno <error number> on fork() for <description>
```
The <error number> argument represents the UNIX error number, and <description> indicates why the process is cloning itself. The task failed to clone itself. This might happen if the maximum number of processes in the system has been reached. To see if this is the case, issue the **ps -eaf | pg** command.

```
(*, **, *C)XXX106 ENV variable <name> undefined
```
The <name> argument represents the name of the undefined environment variable. A variable is undefined in the run-time environment of the task. Check the sym_defs file if there is no entry for the given environment variable. If there is an entry, see if the variable is exported.

```
(*, **, *C)XXX107 ENV variable <name> is undefined, using default value <default string>
```
The <name> argument represents the name of the undefined environment variable, and <default string> represents the default string to be used. A variable is undefined in the run-time environment of the task. The task is using a default string value.

`(*, **, *C)XXX108 ENV variable <name> is undefined, using default value <default string>`
The <name> argument represents the name of the undefined environment variable, and <default string> represents the default string to be used. A variable is undefined in the run-time environment of the task. The task is using a default numeric value.

`(*, **, *C)XXX109 SHM errno <error number> code <return code> getting key for <name>`
The <error number> argument represents the return code of the function that provides the mapping between shared memory names and the shared memory keys, <return code> represents the UNIX error number, and <name> represents the shared memory name.

The task cannot get the shared memory key value for the given shared memory name. Either the acecmid file is not in the files directory or the named shared memory is not defined in the acecmid file. Contact the Cisco Technical Assistance Center to update this file.

`(*, **, *C)XXX110 SHM errno <error number> creating key=$ <memory key>`
The <error number> argument represents the UNIX error number, and <memory key> represents the shared memory key. The task failed to create the shared memory with the given key. A shared memory with the same key might already exist.

`(*, **, *C)XXX111 SHM errno <error number> attaching to key=$ <memory key>`
The <error number> argument represents the UNIX error number, and <memory key> represents the shared memory key. The task failed to attach to the shared memory with the given key. The shared memory with the given key might not yet have been created or might have been removed.

`(*, **, *C)XXX112 SHM chunk not defined for <memory chunk>`
The <memory chunk> argument represents the name of the memory chunk that caused the lookup failure. The task failed to look up a memory chunk in the shared memory region. Check the configuration file in the files directory.

`(*, **, *C)XXX113 SHM chunk is <assigned memory> bytes for <expected memory> Expecting <chunk size>`
The <assigned memory> argument represents the memory chunk size, <expected memory> represents the expected memory chunk size, and <chunk size> represents the memory chunk size in bytes. The expected memory chunk size of the task does not match the allocated chunk. Check the configuration file in the files directory.

`(*, **, *C)XXX114 MSGQ errno <error number> code <return code> getting key for queue name`
The <error number> argument represents the UNIX error number, <return code> represents the return code of the function that provides the mapping between the message queue names and message queue keys, and <queue name> represents the message queue name.

The task cannot get the message queue key value for the given message queue name. Either the acecqid file is not in the files directory or the named message queue is not defined in the acecqid file.

`(*, **, *C)XXX115 MSGQ errno <error number> creating key=$ <queue key>`
The <error number> argument represents the UNIX error number, and <queue key> represents the message queue key. The task cannot create a message queue with the given key. A message queue with the same key might already exist. Look in /CiscoBAMS/files/acecqid for duplicate key definitions. Contact the Cisco Technical Assistance Center to correct the configuration if duplicates are found.

`(*, **, *C)XXX116 MSGQ errno <error number> code <return code> on sendlocal() key=$ <queue key>`
The <error number> argument represents the UNIX error number, <return code> represents the return code of sendlocal(), and <queue key> represents the message queue key.

The task failed to send a message to a task on the same machine. The message queue with the given key might have been removed, or the system-imposed maximum number of pending messages might have been reached. From the UNIX prompt, issue the command **ipcs -qob**. Check if the queue with the given key still exists and if the total QNUM is less than 40.

```
(*, **, *C)XXX117 MSGQ errno <error number> code <return code> on sendremote() key=$<queue
key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of sendremote(), and <queue key> represents the message queue key.

The task failed to send a message to a task on a remote machine. The message queue for the task dealing with intermachine communication might have been removed, or the system-imposed maximum number of pending messages might have been reached. From the UNIX prompt, issue the command **ipcs -qob**. Check if the queue with the given key still exists and if the total QNUM is less than 40.

```
(*, **, *C)XXX118 MSGQ errno <error number> code <error number> on recvmsg() key=$<queue
key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of recvmsg(), and <queue key> represents the message queue key. The task failed to receive a message from a message queue with the given key. The message queue with the given key might have been removed, or the task might have been interrupted.

```
(*, **, *C)XXX119 SEMA errno <error number> code <return code> getting key for <semaphore
name>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of the function that provides the mapping between semaphore names and semaphore keys, and <semaphore name> represents the semaphore name. The task cannot get the semaphore key value for the given semaphore name. Either the acecsid file is not in the files directory, or the named semaphore is not defined in the acecsid file.

```
(*, **, *C)XXX120 SEMA errno <error number> creating key=$<semaphore key>
```
The <error number> argument represents the UNIX error number, and <semaphore key> represents the semaphore key. The task cannot create the semaphore with the given key. A semaphore with the same key might already exist.

```
(*, **, *C)XXX121 FILE errno <error number> opening <name> for <file open>
```
The <error number> argument represents the UNIX error number, <name> represents the name of the file, and <file open> represents the file open modifier (for example, create, read, or append). The task failed to open the named file using the indicated file open operation.

```
(*, **, *C)XXX122 FILE errno <error number> reading <filename> size <total byte size>
```
The <error number> argument represents the UNIX error number, <filename> represents the filename or item being read, and <total byte size> represents the total byte size available to read. The task failed to read the requested number of bytes.

```
(*, **, *C)XXX123 FILE errno <error number> writing <filename> size <total byte size>
```
The <error number> argument represents the UNIX error number, <filename> represents the filename or item being written, and <total byte size> represents the total byte size available to write. The task failed to write the requested number of bytes. Check if the disk is full and if the file exists.

```
(*, **, *C)XXX124 FILE errno <error number> seeking <filename> offset <offset> whence
<flag>.
```
The <error number> argument represents the UNIX error number, <filename> represents the name or description of the input file, <offset> represents the requested offset, and <flag> represents the seek flag (0, 1, or 2). The task failed to move the pointer on the named file.

```
(*, **, *C)XXX125 FILE fprintf() returned <return code>. Expecting <number>.
```
The <return code> argument represents the return code of fprintf(), and <number> represents the number of items requested. The task failed to write the requested number of items.

```
(*, **, *C)XXX126 FILE fscanf() returned <return code>. Expecting <number>
```
The <return code> argument represents the return code of fscanf(), and <number> represents the number of items requested. The task failed to read the requested number of items.

```
(*, **, *C)XXX127 FILE errno <error number> linking <file> to <name>
```
The <error number> argument represents the UNIX error number, <file> represents the file to link from, and <name> represents the name of the new file link. The task failed to create a link to a file. This might happen when the first file does not exist.

```
(*, **, *C)XXX128 FILE errno <error number> unlinking <unlink>.
```
The <error number> argument represents the UNIX error number, and <unlink> represents the name of the file to unlink. The task failed to remove a file link. Check if the named file exists. If it does, check the permissions in the file.

```
(*, **, *C)XXX129 FILE errno <error number> renaming <rename> to <new name>
```
The <error number> argument represents the UNIX error number, <name> represents the original filename, and <new name> represents the new name. The task failed to rename a file.

```
(*, **, *C)XXX130 DIR errno <error number> changing to <directory>.
```
The <error number> argument represents the UNIX error number, and <directory> represents the directory name. The task failed to set its working directory. Check if the directory exists and if the execute permissions are set.

```
(*, **, *C)XXX131 DIR errno <error number> opening <directory>.
```
The <error number> argument represents the UNIX error number, and <directory> represents the directory name. The task failed to open the named directory. Check if the directory exists and if the read permissions are set.

```
(*, **, *C)XXX132 DIR errno <error number> reading <error number>
```
The <error number> argument represents the UNIX error number, and <directory name> represents the directory name. The task failed to read an entry from the named directory. Check if the directory exists and if the read permissions are set.

```
(*, **, *C)XXX133 SCKT errno <error number> creating <protocol> socket for <socket>
```
The <error number> argument represents the UNIX error number, <protocol> represents the protocol (TCP or UDP), and <socket> represents the use of the socket. The task failed to create an end point for communication (a socket). This can happen if the task has opened too many sockets.

```
(*, **, *C)XXX134 SCKT errno <error number> connecting to <address>/<port number>
```
The <error number> argument represents the UNIX error number, <address> represents the server name or address, and <port number> represents the service name or port number (as a string).

The task failed to initiate a connection to a server process. This can be due to one of two things: the server name does not have an entry in the /etc/hosts file, or the service name is not in the /etc/services file. If an IP address and a port number are used instead of names and the message still shows up, contact the Cisco Technical Assistance Center.

```
(*, **, *C)XXX135 SCKT errno <error number> binding socket <socket>
```
The <error number> argument represents the UNIX error number, and <socket> represents the socket descriptor. The task failed to link a name to a socket.

```
(*, **, *C)XXX136 SCKT errno <error number> listening on socket <socket>, qlength=<queue length>
```
The <error number> argument represents the UNIX error number, <socket> represents the socket descriptor, and <queue length> represents the requested queue length. The task failed to listen for connections on a socket.

```
(*, **, *C)XXX137 SCKT errno <error number> accepting from socket=<socket>
```
The <error number> argument represents the UNIX error number, and <socket> represents the socket ID. The task failed to accept a connection from a socket.

```
(*, **, *C)XXX138 SCKT errno <error number> reading <description> size <number>
```
The <error number> argument represents the UNIX error number, <description> represents the description of data to read, and <number> represents the number of bytes to read. The task encountered an error while reading data from a socket.

```
(*, **, *C)XXX139 SCKT errno <error number> writing <description> size <number>
```
The <error number> argument represents the UNIX error number, <description> represents the description of data to write, and <number> represents the number of bytes to write. The task encountered an error while writing data to a socket.

```
(*, **, *C)XXX140 FILE errno <error number> getting file stat for <name>
```
The <error number> argument represents the UNIX error number, and <name> represents the name or description of a file. The task failed to get file statistics. Check if the file exists.

```
(*, **, *C)XXX141 MSGQ errno <error number> code <return code> deleting key=$<queue key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of the routine for deleting the message queue, and <queue key> represents the message queue key. The task failed to delete the message queue with the given key. The message queue might have been deleted manually or by another task.

```
(*, **, *C)XXX142 SHM errno <error number> code <return code> deleting key=$<queue key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of the routine to delete the shared memory, and <queue key> represents the message queue key. The task failed to delete the shared memory with the given key. The shared memory might have been deleted manually or by another task.

```
(*, **, *C)XXX143 SHM errno <error number> attaching to <name> shared memory
```
The <error number> argument represents the UNIX error code, and <name> represents the task name of shared memory owner. This alarm message appears only if the task does not explicitly acquire the shared memory ID through get_mid().

The task failed to attach to the shared memory owned by the given task. The shared memory might not have been created, or it might have been deleted by the owning process. Check if the shared memory owner is executing by issuing the UNIX command **ps -eaf | grep** <task>, where <task> is the owner of the shared memory.

```
(*, **, *C)XXX200 Timed out waiting for <task>
```
The <task> argument represents the description of what the task was waiting for. The task logs the message when the attempt to receive some information from another task or task component has exceeded a certain time threshold.

```
(*, **, *C)XXX201 Invalid command-line option, <switch>
```
The <switch> argument represents the unsupported character command-line switch. The task was passed an unknown or unsupported command-line argument. The configuration file for the task contains an invalid option. To check all available options, invoke the task using the **-h** option and check the contents of the configuration file.

```
(*, **, *C)XXX202 Unrecoverable failure
```
The task cannot recover from an error. Previous or following entries in the log file provide more information on the possible cause of the failure.

```
(*, **, *C)XXX203 SIG errno <error number> setting handler for signal <signal>
```
The <error number> argument represents the UNIX error number, and <signal> represents the signal number. The task failed to set the handler for the given signal. Contact the Cisco Technical Assistance Center if necessary.

```
(*, **, *C)XXX204 SYS errno <error number> on system() for <argument>
```
The <error number> argument represents the UNIX error number, and <argument> represents the description of the argument passed to the system() call.

The task detected an error when passing a command to the UNIX shell. The shell may be unable to find and execute the specified command. Check if the command or script passed to the system call is valid. If it is a script, verify that the execute permissions are set. A system() call might also fail if an excessive number of UNIX processes are already executing. To see if this is the case, issue the command **ps -eaf | pg**.

`(*, **, *C)XXX205 SYS errno <error number> on fork() for <description>`
The <error number> argument represents the UNIX error number, and <description> represents the description of why the process is cloning itself. The task failed to clone itself. This might happen when the maximum number of processes in the system has been reached. To see if this is the case, issue the command **ps -eaf | pg**.

`(*, **, *C)XXX206 ENV variable <name> undefined`
The <name> argument represents the name of the undefined environment variable. A variable is undefined in the run-time environment of the task. Check the sym_defs file to verify there is an entry for the given environment variable. If there is an entry, verify that the variable is exported.

`(*, **, *C)XXX207 ENV variable <name> is undefined, using default value <default string>`
The <name> argument represents the name of the undefined environment variable, and <default string> represents the default string to be used. A variable is undefined in the run-time environment of the task. The task uses a default string value.

`(*, **, *C)XXX208 ENV variable <variable> is undefined, using default value <default string>`
The <variable> argument represents the name of the undefined environment variable, and <default string> represents the default string to be used. A variable is undefined in the run-time environment of the task. The task is using a default numeric value.

`(*, **, *C)XXX209 SHM errno <error number> code <return code> getting key for <memory name>`
The <error number> represents the UNIX error number, <return code> represents the return code of the function that provides the mapping between shared memory names and shared memory keys, and <memory name> represents the shared memory name.

The task cannot get the shared memory key value for the given shared memory name. Either the acecmid file is not in the files directory or the named shared memory is not defined in the acecmid file. Contact the Cisco Technical Assistance Center if necessary.

`(*, **, *C)XXX210 SHM errno <error number> creating key=$<memory key>`
The <error number> argument represents the UNIX error number, and <memory key> represents the shared memory key. The task failed to create the shared memory with the given key. A shared memory with the same key might already exist.

`(*, **, *C)XXX211 SHM errno <error number> attaching to key=$<memory key>`
The <error number> argument represents the UNIX error number, and <memory key> represents the shared memory key. The task failed to attach to the shared memory with the given key. The shared memory with the given key might not yet have been created or it might have been removed.

`(*, **, *C)XXX212 SHM chunk not defined for <lookup failure>`
The <lookup failure> argument represents the name of the memory chunk that caused the lookup failure. The task failed to look up a memory chunk in the shared memory region. Check the configuration file in the files directory.

`(*, **, *C)XXX213 SHM chunk is <assigned> bytes for <name>. Expecting <size> bytes`
The <assigned> argument represents the assigned memory chunk size, <name> represents the memory chunk name, and <size> represents the expected memory chunk size. The expected memory chunk size of the task does not match the allocated chunk. Check the configuration file in the files directory.

`(*, **, *C)XXX214 MSGQ errno <error number> code <return code> getting key for <queue name>`
The <error number> argument represents the UNIX error number, <return code> represents the return code of the function that provides the mapping between message queue names and message queue keys, and <queue name> represents the message queue name.

The task cannot get the message queue key value for the given message queue name. Either the acecqid file is not in the files directory or the named message queue is not defined in the acecqid file.

(*, **, *C)XXX215 MSGQ errno <error number> creating key=$<queue key>

The <error number> argument represents the UNIX error number, and <queue key> represents the message queue key. The task cannot create a message queue with the given key. A message queue with the same key might already exist. Look in the acecqid file found in the /CiscoBAMS/files directory for duplicate key definitions. Contact the Cisco Technical Assistance Center to correct the configuration if you find duplicates.

(*, **, *C)XXX216 MSGQ errno <error number> code <return code> on sendlocal() key=$<queue key>

The <error number> argument represents the UNIX error number, <return code> represents the return code of sendlocal(), and <queue key> represents the message queue key.

The task failed to send a message to a task on the same unit. The message queue with the given key might have been removed, or the system-imposed maximum number of pending messages might have been reached. From the UNIX prompt, issue the command **ipcs -qob**. Check if the queue with the given key still exists and if the total QNUM is less than 40.

(*, **, *C)XXX217 MSGQ errno <error number> code <return code> on sendremote() key=$<queue key>

The <error number> argument represents the UNIX error number, <return code> represents the return code of sendremote(), and <queue key> represents the message queue key.

The task failed to send a message to a task on a remote unit. The message queue for the task dealing with interunit communication might have been removed, or the system-imposed maximum number of pending messages might have been reached. From the UNIX prompt, issue the command **ipcs -qob** and check that the queue with the given key still exists and that the total QNUM is less than 40.

(*, **, *C)XXX218 MSGQ errno <error number> code <return code> on recvmsg() key=$<queue key>

The <error number> argument represents the UNIX error number, <return code> represents the return code of recvmsg(), and <queue key> represents the message queue key. The task failed to receive a message from a message queue with the given key. The message queue with the given key might have been removed, or the task might have been interrupted.

(*, **, *C)XXX219 SEMA errno <error number> code <return code> getting key for <semaphore name>

The <error number> argument represents the UNIX error number, <return code> represents the return code of the function that provides the mapping between semaphore names and semaphore keys, and <semaphore name> represents the semaphore name.

The task cannot get the semaphore key value for the given semaphore name. Either the acecsid file is not in the files directory, or the named semaphore is not defined in the acecsid file.

(*, **, *C)XXX220 SEMA errno <error number> creating key=$<semaphore>

The <error number> argument represents the UNIX error number, and <semaphore> represents the semaphore key. The task cannot create the semaphore with the given key. A semaphore with the same key might already exist.

(*, **, *C)XXX221 FILE errno <error number> opening <name> for <file>

The <error number> argument represents the UNIX error number, <name> represents the name of the file, and <file> represents the file open modifier (for example, create, read, or append). The task failed to open the named file using the indicated file open operation.

(*, **, *C)XXX222 FILE errno <error number> reading <filename> size <byte size>

The <error number> argument represents the UNIX error number, <filename> represents the filename or item being read, and <byte size> represents the total byte size to read. The task failed to read the requested number of bytes.

(*, **, *C)XXX223 FILE errno <error number> writing <byte size> size <filename>

The <error number> argument represents the UNIX error number, <byte size> represents the total byte size to write, and <filename> represents the filename or item being written. The task failed to write the requested number of bytes. Check if the disk is full and if the file exists.

```
(*, **, *C)XXX224 FILE errno <error number> seeking <filename> offset <offset> whence
<flag>
```
The <error number> argument represents the UNIX error number, <filename> represents the name or the description of the input file, <offset> represents the requested offset, and <flag> represents the seek flag (0, 1, or 2). The task failed to move the pointer on the named file.

```
(*, **, *C)XXX225 FILE fprintf() returned <return code>. Expecting <number>
```
The <return code> argument represents the return code of fprintf(), and <number> represents the number of items requested. The task failed to write the requested number of items.

```
(*, **, *C)XXX226 FILE fscanf() returned <return code>. Expecting <number>
```
The <return code> argument represents the return code of fscanf(), and <number> represents the number of items requested. The task failed to read the requested number of items.

```
(*, **, *C)XXX227 FILE errno <error number> linking <file> to <new file>
```
The <error number> argument represents the UNIX error number, <file> represents the file to link from, and <new file> represents the name of the new file link. The task failed to create a link to a file. This might happen when the first file does not exist.

```
(*, **, *C)XXX228 FILE errno <error number> unlinking <name>
```
The <error number> argument represents the UNIX error number, and <name> represents the name of file to unlink. The task failed to remove a file link. Check if the named file exists. If it does, check the permissions in the file.

```
(*, **, *C)XXX229 FILE errno <error number> renaming <name> to <new name>
```
The <error number> argument represents the UNIX error number, <name> represents the original filename, and <new name> represents the new name. The task failed to rename a file.

```
(*, **, *C)XXX230 DIR errno <error number> changing to <directory name>
```
The <error number> argument represents the UNIX error number, and <directory name> represents the directory name. The task failed to set its working directory. Check if the directory exists and if the execute permissions are set.

```
(*, **, *C)XXX231 DIR errno <error number> opening <directory name>
```
The <error number> argument represents the UNIX error number, and <directory name> represents the directory name. The task failed to open the named directory. Check if the directory exists and if the read permissions are set.

```
(*, **, *C)XXX232 DIR errno <error number> reading <directory name>
```
The <error number> argument represents the UNIX error number, and <directory name> represents the directory name. The task failed to read an entry from the named directory. Check if the directory exists and if the read permissions are set.

```
(*, **, *C)XXX233 SCKT errno <error number> creating <protocol> socket for <use>
```
The <error number> argument represents the UNIX error number, <protocol> represents the protocol (TCP or UDP), and <use> represents the use of the socket. The task failed to create an end point for communication (a socket). This can happen if the task has opened too many sockets.

```
(*, **, *C)XXX234 SCKT errno <error number> connecting to <name>/<string>
```
The <error number> argument represents the UNIX error number, <name> represents the server name or address, and <string> represents the service name or port number (as a string).

The task failed to initiate a connection to a server process. Either the server name does not have an entry in the /etc/hosts file, or the service name is not in the /etc/services file. If an IP address and a port number are used instead of names and the message still shows up, contact the Cisco Technical Assistance Center.

```
(*, **, *C)XXX235 SCKT errno <error number> binding socket <descriptor>
```
The <error number> argument represents the UNIX error number, and <descriptor> represents the socket descriptor. The task failed to bind a name to a socket.

```
(*, **, *C)XXX236 SCKT errno <error number> listening on socket <descriptor>,
qlength=<queue length>
```
The <error number> argument represents the UNIX error number, <descriptor> represents the socket descriptor, and <queue length> represents the requested queue length. The task failed to listen for connections on a socket.

```
(*, **, *C)XXX237 SCKT errno <error number> accepting from socket=<id>
```
The <error number> argument represents the UNIX error number, and <id> represents the socket ID. The task failed to accept a connection from a socket.

```
(*, **, *C)XXX238 SCKT errno <error number> reading <data> size <bytes>
```
The <error number> argument represents the UNIX error number, <data> represents the description of data to read, and <bytes> represents the number of bytes to read. The task encountered an error while reading data from a socket.

```
(*, **, *C)XXX239 SCKT errno <error number> writing <data> size <bytes>
```
The <error number> argument represents the UNIX error number, <data> represents the description of data to write, and <bytes> represents the number of bytes to write. The task encountered an error while writing data to a socket.

```
(*, **, *C)XXX240 FILE errno <error number> getting file stat for <description>
```
The <error number> argument represents the UNIX error number, and <description> represents the name or description of a file. The task failed to get file statistics. Check if the file exists.

```
(*, **, *C)XXX241 MSGQ errno <error number> code <return code> deleting key=$<queue key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of the routine for deleting the message queue, and <queue key> represents the message queue key. The task failed to delete the message queue with the given key. The message queue might have been deleted manually or by another task.

```
(*, **, *C)XXX242 (SHM errno <error number> code <return code> deleting key=$<queue key>
```
The <error number> argument represents the UNIX error number, <return code> represents the return code of the routine for deleting the shared memory, and <queue key> represents the message queue key. The task failed to delete the shared memory with the given key. The message queue might have been deleted manually or by another task.

```
(*, **, *C)XXX243 SHM errno <error code> attaching to <task name> shared memory
```
The <error code> argument represents the UNIX error code, and <task name> represents the task name of the shared memory owner.

The task failed to attach to the shared memory owned by the given task. The shared memory might not have been created or might have been deleted by the owning process. Check if the shared memory owner is executing by issuing the UNIX command **ps -eaf | grep** <task>, where <task> is the owner of the shared memory.

```
(*, **, *C)XXX244 SYS errno <error code> on UNIX system call <call>
```
The <error code> represents the UNIX error code, and <call> represents the system call name. The task encountered an error during a UNIX system call.

Manager Task-Specific Messages

The Manager (MGR) task-specific error messages are numbered from 400 through 899 (not all numbers are used). The specific tasks use the same range of numbers, but the text of the message is different for each task.

MGR500 Spawning <task>
The <task> argument represents the task name. The MGR task manages the BAMS software. The MGR task starts and stops the other BAMS tasks. This informational message indicates that a task is being started.

MGR501 Sending <signal number> to <task>
The <signal number> argument represents the signal number, and <task> represents the task name. Some communication between UNIX tasks is by signal number. This message indicates that a signal has been sent to a UNIX task.

MGR502 Attempting a restart
This message indicates that the MGR task is attempting to restart the system. The startup messages that follow this message are displayed and are logged in the syslog file.

MGR503 Keepalive internal passed
Keep alive is a security feature that monitors system activity when you log on to the system as an administrator or operator. If control session activity is idle for more than a desired amount of time, the application logs the user out. This message indicates that the keep alive interval has passed.

MGR504 Task <name> will be restarted
The <name> argument represents the task name. The MGR is restarting the displayed task.

MGR505 Invoking task <name>
The <name> argument represents the task name. The MGR is starting the displayed task.

* MGR600 Function <name> failed, code <rc>, errno:<UNIX error number>
The <name> argument represents the function name, <rc> represents the return code, and <UNIX error number> represents the UNIX error number. The displayed function returned an unexpected return code.

* MGR601 Too many options for <process>
The <process> argument represents the process name. This message indicates that too many optional parameters are passed to a process listed in the proglist file. The proglist file contains the names of configuration files specific to each task.

Verify the parameters in the configuration file of the offending task. If the configuration file has been modified, remove some of the options within the configuration file or restore the original configuration file.

** MGR700 Too few parameters <total>/<expected> on line <number> of proglist
The <total> argument represents the number of parameters, <expected> represents the number of expected parameters, and <number> represents the line number in the file. The proglist file has an incorrect number of parameters. Contact the Cisco Technical Assistance Center.

** MGR800 Failed spawning <name>
The <name> argument represents the task name. This message indicates that the MGR task is unable to start the displayed task. This message can be caused by a combination of hardware and software problems.

Restart the application software and monitor the system log for the same occurrence. If the problem persists, restart the operating system. If the problem is still present, reinstall the application software.

```
*C MGR801 Task <name> has died.
```
The <name> argument represents the task name. This message indicates that the specified task has stopped responding to the system. The MGR task automatically attempts a restart of the indicated task. Verify the permissions and ownerships on the task that has died. Restart the application and monitor the system log to see if the error persists. If MGR is unable to restart the task, you might need to reinstall the software. To clear this alarm, you must provide a matching text message because multiple MGR801 alarms may be active (each with a different message).

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
*C MGR802 Manager shutting down the application
```
This message indicates that a task has exited due to the receipt of the specified UNIX signal. The application has been terminated with a software interrupt. A signal 15 indicates that the system received the **stop** command. A signal 11 indicates that a segmentation violation occurred. The MGR task attempts a restart of the application for signals other than signal 15. If the MGR task is unable to restart the application, a UNIX shutdown might be necessary. This error can be associated with a hardware or a software failure.

```
*C MGR803 Received unexpected signal number <number>. Exiting
```
The <number> argument represents the signal number. This error message indicates that the task has exited due to the receipt of the specified UNIX signal. The MGR task should reinitialize the system automatically. Monitor the syslog file during the next hour. If the error message reoccurs, reinstall the software.

```
*C MGR804 Disk full. Must be cleaned up manually
```
The system has a full data directory and the Mass Storage Control task is unable to effect an automatic system cleanup. This type of error can occur if the disk thresholds are set too high.

Change to the /CiscoBAMS/data directory and remove older data files. The Mass Storage Control parameters should be verified and changed as necessary. For more information about setting up or editing MSC parameters, see the "Setting Up Disk Monitoring Thresholds" section on page 2-18.

Accumulation Task Messages

The Accumulation (ACC) task accumulates peg counts for measurements. During processing, this task writes messages to the syslog with an ACC task identifier (for example, ACC001).

```
ACC000 <name> version <number1> has started, pid <number2>
```
The <name> argument represents the task name. The <number1> argument represents the task version. The <number2> argument represents the process ID number. This message indicates that the ACC task has started. This is an informational message. No action is necessary.

```
ACC001 <name> terminating with code <errno>
```
The <name> argument represents the task name. The <errno> argument represents the UNIX error number set by the system function. This message indicates that the ACC task has been ended. See the error number to determine the reason for the error.

```
ACC002 <name> terminating with signal number <number>
```
The <name> argument represents the task name. The <number> argument represents the signal number. This message indicates that the ACC task has been ended. See the signal number to determine the reason for the error.

```
ACC003 <name> terminating normally
```
The <name> argument represents the task name. This message indicates that the ACC task is ending normally. This is an informational message. No action is necessary.

ACC004 Processing file <name>

The <name> argument represents the filename. This message indicates that the ACC task is processing the displayed file. This is an informational message. No action is necessary.

ACC005 File <name> processed

The <name> argument represents the filename. This message indicates that the ACC task processed the displayed file. This is an informational message. No action is necessary.

ACC010 <number> calls processed

The <number> argument represents the number of processed calls. This message indicates the number of call records that the ACC task has processed. This is an informational message. No action is necessary.

* ACC019 No flat files to process. Retry in <number> seconds

The <number> argument represents the number of seconds. This message indicates that the ACC task has found no data files to process and is pausing for a period of time before attempting to process additional files. If no *, **, or *C is displayed, this message is displayed as an informational message. No action is necessary.

ACC022 Processed call set in <number> seconds

The <number> argument represents the number of seconds. This message indicates the elapsed time to process a single call set (data file). This is an informational message. No action is necessary.

ACC023 <name>: User time=<time1> sec, CPU time=<time2> sec

The <name> argument represents the process name. The <time1> argument represents the CPU time used for executing instructions in user space. The <time2> argument represents the CPU time used for executing instructions in system space. This alarm message indicates the user and CPU time used for processing enclosure files. This is an informational message. No action is necessary.

* ACC101 File error: <name> on <filename> (<errno>)

The <name> argument represents the name of the failed system call. The <filename> argument represents the filename. The <errno> argument represents the UNIX error number set by the system function. This message indicates that the ACC task encountered an error when performing a system call operation on the specified file. See the UNIX error number for more information.

*C ACC102 File I/O error: <name> on <filename>, requested: <number1>, got: <number2> (<errno>)

The <name> argument represents the name of the failed system call. The <filename> argument represents the filename. The <number1> argument represents the number of requested bytes. The <number2> argument represents the number of received bytes. The <errno> argument represents the UNIX error number set by the system function. This message indicates that the ACC task encountered an error when performing a file I/O operation. Contact the Cisco Technical Assistance Center.

W ACC204 Duplicate Trunk Group Key <trunk group> IC:<carrier id>

The <trunk group> argument represents the trunk group. The <carrier id> argument represents the carrier identifier. An error occurred when the trunk group key was added. Contact the Cisco Technical Assistance Center.

** ACC206 Undefined environment variable <varname>

The <varname> argument represents the variable name. This message indicates that an expected environment variable has not been defined. Try logging out and logging back in. If this problem persists, verify that the specified environment variable is defined in /CiscoBAMS/bin/sym_defs. If it is missing, contact the Cisco Technical Assistance Center for directions on how to proceed.

** ACC207 Function <name> failed <errno>

The <name> argument represents the name of the failed function. The <errno> argument represents the UNIX error number set by the system function. This message indicates that an unexpected function failed. Contact the Cisco Technical Assistance Center.

*C ACC210 The EOF marker missing for the file <filename>

The <filename> argument represents the filename. This message indicates a missing end-of-file marker in the displayed file. Contact the Cisco Technical Assistance Center.

`* ACC221 Old call set received <id> current <interval string>`
The <id> argument represents a new call set, and <interval string> represents the current call set. This message appears when a call set older than the current one is received. Contact the Cisco Technical Assistance Center.

`* ACC222 Missing call set expected <next interval string> received <call set id>`
The <next interval string> argument represents a new call set, and <call set id> represents the current call set. This message appears when a time gap has been detected in the ACC input stream. Contact the Cisco Technical Assistance Center.

`* ACC223 Can't load interval value from check point file`
This message appears when an error is detected during the reading of the checkpoint file. Contact the Cisco Technical Assistance Center.

`* ACC224 Memory allocation failed requested by <function name>`
The <function name> argument represents the name of the function requesting memory. Contact the Cisco Technical Assistance Center. This message appears when memory is allocated and none is returned.

`ACC225 No Checkpoint file to load`
This message appears when the ACC task starts and no checkpoint file is found. This is an informational message. No action is necessary.

`ACC226 Check point loaded, last interval was <current interval string>`
The <current interval string> argument represents the current interval string loaded from the checkpoint. This message appears when the ACC task starts and the checkpoint file is loaded. This is an informational message. No action is necessary.

`* ACC227 <count direction> <description>, TG <trunk group prefix> <trunk group number>`
`<counter value> <condition> <threshold value> <timestamp>`
The <count direction> represents the direction of the count, <description> is a description of the count, <trunk group prefix> represents the trunk group prefix, <trunk group> represents the trunk group, <counter value> represents the counter value, <condition> represents the threshold crossing condition, <threshold value> represents the value of the threshold, and <timestamp> represents the timestamp (that is, `@ YYYYMMDDHHMMSS UTC`). This message appears when a counter passes a user-determined threshold test or clears. To clear this alarm, you must provide a matching text message, because multiple ACC227 alarms might be active (each with a different message).

> **Note**   This alarm is cleared after the alarm condition is removed.

`ACC228 <count direction> <description>, TG <trunk group number> contains an invalid`
`condition`
The <count direction> represents the direction of the count, <description> is a description of the count, and <trunk group number> represents the trunk group number. This message appears when a threshold is tested and the condition (<=>) is an unknown test. This is an informational message. No further action is necessary.

`* ACC230 CTL file error: <string>`
The <string> argument represents a string from ai_strerror(). This message displays errors from calls to the Web Application Program Interface (API). Contact the Cisco Technical Assistance Center.

`* ACC231 Threshold file error <key> <threshold name> <value> <text>`
The <key> argument represents the id/tag, <threshold name> represents the name of the threshold (for example, "ANSWERED"), <value> represents the value in error, and <text> represents a description of the error condition. This message appears when the threshold file contains an incorrect value. Contact the Cisco Technical Assistance Center.

* ACC232 Multiple Global thresholds found
On each new call set, ACC reloads the Threshold Crossing Alarms table if the file contains more than one global threshold record that this alarm displays. Contact the Cisco Technical Assistance Center.

* ACC233 Out of table Space
If ACC attempts to insert a new node in a tree and no space is available, this alarm is displayed. Contact the Cisco Technical Assistance Center.

* ACC234 Known node can't be found deleting
If ACC attempts to delete a node in a tree and the delete returns null (node not found), this alarm is displayed. Contact the Cisco Technical Assistance Center.

* ACC235 CTL file <name> value error: <value>
The <name> argument represents the name of the CTL file and the value represents the measurement value. This message displays when the value of the measurement field in the control file is not 0 or 1. Check the configuration file using MML and fix the value.

* ACC236 Interval minutes must be 5, 10, 15, 20 or 30
This message displays when the ACC task reads an invalid value from the NODEPARMS configuration file for the interval minutes. Check the NODEPARMS configuration file using MML and fix the value.

* ACC237 Number of Ckts not populated for TD <trunk group>
The <trunk group> number represents the trunk group number. This message displays if the number of circuits for a trunk group has not been defined or has been set to zero. An invalid zero value leads to invalid statistics. If this alarm is generated when the number of circuits is zero, the percentage measurements are reported as 100 percent. At the node level, modify the Trunk Group table (TRUNKGRP tag ID) for the specified trunk group and then assign a non-zero value to the circuits parameter.

Augmentation BAF Task Messages

The Augmentation BAF (AGB) task controls the augmentation of data through the use of lookup tables. During processing, this task writes messages to the syslog with an AGB (Augmented BAF) task identifier (for example, AGB001).

AGB000 <task> version <version> has started <pid>
The <task> argument represent the task name, <version> represents the task version, and <pid> represents the process identifier. This message indicates that the AUG task has started. This is an informational message. No action is necessary.

AGB001 <task> terminating with <code> <pid>
The <task> argument represents the task name, <code> represents UNIX error code, and <pid> represents the process identifier. This message indicates that the AUG task has ended abnormally. See the UNIX error code for an explanation of the problem.

AGB002 <name> terminating with signal <number1>, pid <number2>
The <name> argument represents the task name, <number1> represents the UNIX signal number, and <number2> represents the UNIX process identifier. This message indicates that the displayed signal ended the task. See the UNIX signal number for an explanation of the problem.

AGB003 <name> terminating normally, pid <number>
The <name> argument identifies the task name, and <number> identifies the process ID number of the task. This message indicates that the task terminated normally. This is an informational message. No action is necessary.

AGB004 Processing file <name>
The <name> argument represents the filename. This message indicates that the AUG task is processing the displayed file. This is an informational message. No action is necessary.

AGB009 Received system signal <signal name>.
The <signal name> argument represents the name of the UNIX signal. The AUG task received a system signal. This is an informational message. No action is necessary.

AGB010 Processed <filename>, total: <number> good: <number> sunerr: <number> semerr <number> lk uperr <number>
The <filename> argument represents the name of the processed file, and <number> represents the total number of call records, the number of good call records, the number of call records with syntax errors, the number of call records with semantic errors, and the number of call records with error lookups, respectively. This message provides summary information of call records being processed. This is an informational message. No action is necessary.

AGB019 No flat files to process. Retry in <number> seconds.
The <number> argument represents the time delay in seconds. There are no data files to process. The task waits and retries in the specified number of seconds. This is an informational message. No action is necessary.

AGB024 Creating <name> table
The <name> argument represents the lookup table name (that is, NPANXX table, Rating Exception table, Rating Type table, or Trunk Group table). The named lookup table is created and loaded into memory. This is an informational message. No action is necessary.

AGB025 Refreshing <name> table.
The <name> argument represents the lookup table name. The lookup table is reloaded. This is an informational message. No action is necessary.

* AGB100 System call failure: <name> <errno>
The <name> argument represents the name of a failed system call, and <errno> represents the UNIX error number. This message indicates that the AUG task encountered an error when requesting the displayed system call. Look up the UNIX error number. Contact the Cisco Technical Assistance Center.

* AGB101 File error: <call> on <file> (<code>)
The <call> argument represents the name of the failed system call, <file> represents the name of the file being operated on, and <code> is the UNIX error number. This message indicates an error during the performing of the displayed system call operation. Look up the UNIX error number. A software-related problem is indicated. Contact the Cisco Technical Assistance Center.

*C AGB102 File I/O error: <call> on <file>, requested <number1>, got <number2>
The <call> argument represents the failed system call, <file> represents the name of the file being operated on, <number1> represents the number of bytes requested, and <number2> represents the number of bytes received. This message indicates an error during performance of the displayed I/O file operation. Check if the disk is full or not. If the disk is not full, a software-related problem is indicated. Contact the Cisco Technical Assistance Center.

* AGB104 Cannot find file <name> to process
The <name> argument represents the filename. This message indicates that the task is expected to reprocess the named file but did not find it. Contact the Cisco Technical Assistance Center.

* AGB201 Undefined environment variable <name>
The <name> argument represents the name of the undefined environment variable. This message indicates that an environment variable is not defined. Check the sym_defs file. Contact the Cisco Technical Assistance Center.

* AGB202 <index> - Invalid <field>: <value>
The <index> argument represents the call record index, <field> represents the field name in the call record, and <value> indicates the value of the field. This message indicates that the AUG task detected an invalid field value in the call record. The input data has an invalid call reference ID. Check the Cisco MGC raw data, and make sure the call reference ID has the correct format. Contact the Cisco Technical Assistance Center.

* AGB203 <index> - fail <function> lookup of <field>: <value>
The <index> argument represents the call record index, <function> represents the function name of the table lookup, <field> represents the field name in the call record, and <value> represents the value of the field being looked up. This message indicates a failed table lookup. Make sure the specific value exists in the table. Contact the Cisco Technical Assistance Center.

* AGB210 CTL file: <name> error: <ai string error>
The <name> argument represents the function name, and <ai string error> represents the error string. This message indicates an error returned by a control file function. Make sure the CTL file exists in the specified directory. Check the existence of the CTL file (*.CTL) and verify the structure of the control file (*.CFDT). If the CFDT file is corrupt, copy the default CFDT file. Contact the Cisco Technical Assistance Center.

* AGB211 CTL file BAF value error: <value>
The <value> argument represents a bad value. This message indicates that the content of the BAF field in the control file is not 0 or 1. Verify sysparm configurations. Make sure the BAF output is set to 0 (disabled) or 1 (enabled). Contact the Cisco Technical Assistance Center.

* AGB232 #<index> Fail <function> lookup of<field1>:<value1> <field2>:<value2> <field3>:<value3>
The <index> argument represents the call record index, <function> represents the function name of the table lookup, <field1> represents the first field name in the call record, <value1> represents the value of the first field, <field 2> represents the second field name in the call record, <value2> represents the value of the second field, <field3> represents the third field name in the call record, and <value3> represents the value of the third field. This message indicates a failed table lookup at the named locations. The system tried to match three field values to the lookup table. Verify the table configuration file, and make sure the field values exist in that configuration file. Contact the Cisco Technical Assistance Center.

* AGB233 #<index> Fail <function> lookup of <field>:<value>
The <index> argument represents the call record index, <function> represents the function name, <field> represents the field name of the call record, and <value> represents the value of the field being looked up. This message indicates a failed table lookup of this field. Verify that the input raw file used only one field value to match to the lookup table. Contact the Cisco Technical Assistance Center.

* AGB234 #<index> - Invalid block: <field name>
The <index> argument represents the record index, and <field name> represents field names and values. This message indicates that an invalid block format was encountered in the reading of blocks from the table file. For example, the system expected a six-digit value for NPANXX, but it received only five digits. Verify the field value in the table. Make sure you enter the required value. For field names and descriptions, see the "Tag IDs and Field Names" section on page 5-1.

* AGB235 #<number> records failed trunk group lookup for file cor_agcdr_<timestamp>_<sequence number>.bin
The <number> argument represents the number of records that failed the trunk group lookup, the <timestamp> argument represents the date and time, and the <sequence number> argument represents the file sequence number. This minor alarm indicates that the Trunk Group table is not correctly or fully populated. This error precludes these records from being processed by the measurements function. Fix the Trunk Group table. Refer to the AUG*log files in the opt/CiscoBAMS/files directory. This file specifies which trunk group should be modified so that these configuration failures do not occur.

AGB236 <number> record(s) default to interlata due to absent NPANXX for file
The <number> argument represents the record number that defaulted to the INTERLATA rating type due to this NPANXX combination not being in the NPANXX tables. This message indicates that the AUG task has started. This is an informational message. No action is necessary.

AGB237 <number> record(s) did not have called and or calling number for files <name>.
The <number> argument represents the number of records and the <name argument> represents the filenames of no-data files encountered during processing. This is an informational message. No action is required.

ASCII Conversion Task Messages

The ASCII Conversion (ASC) task outputs Cisco MGC End of Call 1110 ASCII records. During processing, this task writes messages to the syslog with an ASC task identifier (for example, ASC001).

ASC000 <task name> version <version> has started pid <process identifier>
The <task name> argument represents the task name, <version> represents the version, and <process identifier> is the UNIX process identifier. This alarm defines what version of the task is running. This is an informational message. No action is required.

ASC001 <task name> terminating with code <return error code>, pid <process id>
The <task name> argument represents the task name, <return error code> represents the error code returned, and <process id> represents the UNIX process identifier. The named task has been terminated abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

ASC002 <task name> terminating with signal <signal number>, pid <process id>
The <task name> argument represents the task name, <signal number> represents the UNIX signal number, and <process id> is the UNIX process identifier. The named task has been ended by a UNIX signal. This is an informational message. No action is necessary.

ASC003 <task name> terminating normally, pid <process id>
The <task name> argument represents the task name, and <process id> is the UNIX process identifier. The ASC task has terminated normally. This is an informational alarm that appears only in test mode.

ASC004 No file to be processed. Retries in <number> seconds
The <number> argument represents the number of seconds the system waits before reattempting to process files. The task sleeps for the number of seconds displayed before accessing more files. This is an informational message. No action is necessary.

* ASC101 File error: <data format type> on <system call> <errno>
The <data format type> represents the type of data format; <system call> represents the name of the failed system call or the name of the file being operated on; and <errno> represents the UNIX error number. An error was encountered during the performance of a system call operation. Look up the UNIX error number in the listing at the end of this appendix and contact the Cisco Technical Assistance Center.

** ASC200 Environment variable <name> not found
The <name> argument represents the name of the missing environment variable, for example CONFIG_DIR, FILES_DIR, dir_in or dir_out. The displayed environment variable was not found. The symbol definition file (sym_defs) is missing a variable, or the variable must be set before the task is run. Contact the Cisco Technical Assistance Center.

(*,**) ASC201 Function <name> failed, <errno>
The <name> argument represents the calling function name, and <errno> represents the UNIX error number. This message indicates failure of the displayed function call, that is, Major Events for open_file, stat (sysParm.CTL), ai_open ctlfile, minor events for ai_get_max_num_of_recs(), ai_load_rec(), or ai_close_file. Examine the error number. Contact the Cisco Technical Assistance Center.

(*,**) ASC201 Function <calling function> failed, (<error number>)
The <calling function> represents the calling function, and the <error number> argument represents the UNIX error number. The system encountered a failure trying to make a function call. Examine the system error. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

** ASC202 Invalid AMADEF format, read <error string>, expected <expected string>
The <error string> argument represents the error string found, and the <expected string> argument represents the string that was expected. The system encountered a format problem while trying to read the AMADEF file. The AMADEF file needs to be fixed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

* ASC205 Unsupported field <field name>, conversion not defined
The <field name> argument represents the name of the field. The system encountered an AMA field that cannot be processed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

ASC207 Processed <filename>, converted <num recs>, output = <suffix>
The <filename> argument represents the input filename, <num recs> represents the number of records converted, and <suffix> represents the output file suffix. This message displays processing status information. This is an informational message. No action is necessary.

(*,**) ASC208 <data type> code <structure code> not found, cannot process record
The <data type> argument represents the type of data and <structure code> represents the structure code or call type. The system is unable to process a record due to missing code and structure code information. Check the AMADEF file. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

** ASC209 Error <return value> reading file: CALLSTRUCT_MAP
The <return value> represents the value returned. The system encountered a format error reading the callstruct file. Check the CALLSTRUCT_MAP file. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

1110 Binary Task Messages

The 1110 Binary (BIN) task correlates 1010, 1030, 1040, and 1060 CDBs from the Cisco MGC, and generates 1110, and 1060 CDBs. During processing, this task writes messages to the syslog with a BIN task identifier (for example, BIN001).

BIN000 <task> version <version number> has started, pid <process ID>.
The <task> argument represents the task name, <version number> represents the task version number, and <process ID> represents the UNIX process identifier. This message identifies the version of the task that is running. This is an informational message. No action is necessary.

BIN001 <task> terminating with code <return code>, pid <process ID>.
The <task> argument represents the task name, <return code> identifies the terminating signal number encountered by the system, and <process ID> represents the UNIX process identifier. The named task ended abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

BIN002 <task> terminating with signal <number>, pid <process ID>.
The <task> argument represents the task name, <number> represents the UNIX signal number, and <process ID> represents the UNIX process identifier. The named task was ended by a UNIX signal.

BIN003 <task> terminating normally, pid <process ID>.
The <task> argument represents the task name, and <process ID> represents the UNIX process identifier. The named task ended normally. This is an informational message. No action is necessary.

BIN004 Processing file <filename>
The <filename> represents the name of the file being processed. The task has started processing the named file. This is an informational message. No action is necessary.

BIN009 Received system signal <signal>
The <signal> argument represents the received signal number. The BIN task received a system signal. This is an informational message. No action is necessary.

BIN010 Processed <filename>, IN(total: <number1>) OUT(1060: <number2> 1110: <number3>)
The <filename> argument represents the processed filename, <number1> represents the total number of processed call records, <number2> represents the number of 1060 call records, and <number3> represents the number of 1110 call records. This message displays call process summary information. This is an informational message. No action is necessary.

BIN011 Processed <filename>, Skipped CDEs: <number1> Unknown CDEs: <number2>
The <filename> argument represents the processed filename, <number1> represents the number of skipped CDEs, and <number2> represents the number of unknown CDEs. This message indicates the number of CDEs that were skipped or unknown in a processed file. This is an informational message. No action is necessary.

BIN012 Output file <filename>
The <filename> argument represents the processed filename. This message indicates the name of the output file. This is an informational message. No action is necessary.

BIN019 No flat files to process. Retry in <number> seconds.
The <number> argument represents the number of seconds. This message indicates that no data exists for processing. The BIN task retries in the displayed number of seconds. This is an informational message. No action is necessary.

BIN020 CDB tag <number>, CDB is skipped.
The <number> argument represents the CDB number. This CDB is skipped when the CDB tag is found in the Skip CDB table. This is an informational message. No action is necessary.

BIN021 CDE tag <number>, CDB is skipped
The <number> argument represents the CDE number. This CDE is skipped when the CDE tag is found in the Skip CDE table. This is an informational message. No action is necessary.

BIN024 Creating <name> table
The <name> argument represents the lookup table name (that is, NPANXX table, Rating Exception table, Rating Type table, or Trunk Group table). The named lookup table is created and loaded into memory. This is an informational message. No action is necessary.

BIN025 Refreshing <name> table
The <name> argument represents the lookup table name. The lookup table is reloaded. This is an informational message. No action is necessary.

BIN026 Generated CDB <type> length is <length>.
The <type> argument represents the CDB type. The <length> argument represents the length of the CDB. This is an informational message. No action is necessary.

* BIN100 System call failure: <call name> (<error number>)
The <call name> represents the name of the failed system call, and <error number> represents the UNIX error number. An error was encountered when this system call was requested. Look up the UNIX error number for more information. Contact the Cisco Technical Assistance Center.

* BIN101 File error: <call> on <file> (<code>)
The <call> argument represents the name of the failed system call, <file> represents the filename, and <code> represents the UNIX error number. This message indicates that the BIN task encountered an error when performing the displayed system call operation. Look up the error number. Contact the Cisco Technical Assistance Center.

*C BIN102 File I/O error: <call> on <file>, requested: <number1>, got: <number2> (code)
The <call> argument represents the name of the failed I/O system call, <file> represents the name of the file being operated on, <number1> represents the number of bytes requested, and <number2> represents the number of bytes received, and <code> represents the UNIX error number. The BIN task encountered an error when performing the displayed I/O file operation. Contact the Cisco Technical Assistance Center.

* BIN201 Undefined environment variable <name>
The <name> represents the name of the environment variable. This message indicates that the displayed environment variable is not defined. Contact the Cisco Technical Assistance Center.

* BIN202 Input file does not start with header CDB
This message indicates that the BIN task detected an invalid field value in the call record. The BIN task placed the input file in the error directory. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* BIN203 Error reading TAG/LEN of TLV errno=<error number>
The <error number> argument represents the UNIX error number. An error occurred during the reading of the TLV (T = Tag field code, L = Length of the field value, V = field Value). The BIN task stops processing this file, renames it as an error file, and begins processing the next input file. Look up the error. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* BIN204 Input file is not ended by footer CDB
The system encountered the displayed error. The BIN task placed the input file in the error directory. Contact the Cisco Technical Assistance Center.

* BIN207 #<index> - <name>: <value> too long, <size> expected.
The <index> argument represents the call record index, the <name> argument represents the field name, the <value> argument represents the field value, and the <size> argument represents the expected field size. This message indicates that the field string exceeded the expected size. Contact the Cisco Technical Assistance Center.

* BIN210 CTL file: <name>: error: <ai string error>.
The <name> argument represents the control filename, and the <ai string error> argument represents the displayed error string. This message indicates the error returned by a control file function. Contact the Cisco Technical Assistance Center.

* BIN211 CTL file lookup value error: <value>
The <value> argument represents the measurement value. This message displays when the value of the measurement field in the control file is not 0 or 1. Check the configuration file using MML and fix the value.

* BIN222 Invalid TLV len:<length> at <position>th chars, CDB len:<size>
The <length> argument represents the incorrect TLV length, <position> represents the TLV position in the CDB, and <size> represents the size of the CDB. The system read an invalid TLV length. The BIN task stops further processing of the input file and renames the input file by prepending "err_" to the input filename. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* BIN223 #<format> - Get <number1> byte record. <number2> bytes record expected.
The <format> argument represents the data format, <number1> represents the size of the call record read, and <number2> represents the size of the call record expected. This message indicates the processed record size.

* BIN224 Process all CDEs: Error in parsing record.
This message indicates that the BIN task encountered an error during the parsing of this record. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* BIN225 Dump_cde: Unrecognized tag <tag number>
The <tag number> represents the number of the tag. This message indicates that the BIN task encountered an unrecognized tag. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* BIN226 Incorrect field length <field value length>.
The <field value length> argument represents the length of the field value.This message indicates that the BIN task encountered an incorrect field length. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

BIN232 <index> - Fail <function> lookup of <field1>: <value1> and <field2>: <value2>
The <index> argument represents the index of the call record, the <function> argument represents the function name of the table lookup, the <field1> argument represents the first field name in the call record, the <value1> argument represents the first value of the field being looked up, the <field2> argument represents the second field name in the call record, and the <value2> argument represents the second value of the field being looked up. This message indicates that the task failed to look up the indicated fields.

```
BIN233 <index> - Fail <function> lookup of <field>: <value>
```
The <index> argument represents the index of the call record, the <function> argument represents the function name of the table lookup, the <field> argument represents the first field name in the call record, and the <value> argument represents the first value of the field being looked up. This message indicates that the task failed to look up the indicated fields.

```
BIN234 #<record index> - Invalid record <field names/values>
```
The <record index> argument represents the record index, and the <field names/values> argument represents the field names and values. The BIN task encountered an invalid block format while reading blocks from a control file. The message shows which fields are invalid. Use MML to correct the values.

```
BIN235 <number> record(s) failed <trunk/sigpath> lookup for file <name>
```
The <number> argument represents the number of records, the <trunk/sigpath> argument represents the trunk group or sigpath, and the <name> argument represents the filename. This message shows the number of lookup failures per file. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/BIN_cdr.log and search for message BIN233. Correct the appropriate record in the Trunk Group table or in the Nailed Connection table.

```
BIN236 Setting BIN processing prefix to <prefix>
```
The <prefix> argument represents the file prefix. The BIN task processes files whose prefixes match the file-prefix1 parameter in the Poll table. If this parameter is not populated, the file-prefix2 parameter is used. If both parameters are empty, the default prefix is cdr. Whenever the Poll table is modified, this message displays the value of the file prefix that BIN uses.

Correlation Task Messages

The Correlation (COR) task correlates call record data collected from the Cisco MGC. During processing, this task writes messages to the syslog with a COR task identifier (for example, COR001).

```
COR000 <task> version <version number> has started pid <process ID>.
```
The <task> argument represents the version task, <version number> represents the task version number, and <process ID> represents the UNIX process identifier. This message defines the version of the task that is running. This is an informational message. No action is necessary.

```
COR001 <task> terminating with code <return code>, pid <process ID>.
```
The <task> argument represents the task name, <return code> represents the return code, and <process ID> represents the UNIX process identifier. The named task has ended abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

```
COR002 <task> terminating with signal <number>, pid <process ID>.
```
The <task> argument represents the task name, <signal number> represents the signal number received, and <process ID> is the UNIX process identifier. The named task was ended by a UNIX signal. This is an informational message. No action is necessary.

```
COR003 <task> terminating normally, pid <process ID>.
```
The <task> argument represents the task, and <process ID> represents the UNIX process identifier. The named task has ended normally. This is an informational message. No action is necessary.

```
COR004 No file to be processed. Retries in <number> seconds.
```
The <number> argument represents the interval, in seconds, between retries—that is, the task sleeps for <number> seconds before accessing more files. This is an informational message. No action is necessary.

```
COR005 Start processing file <name>.
```
The <name> argument represents the filename. This message indicates that a file is processing. This is an informational message. No action is necessary.

```
COR006 Processed file <name>: Output-> Complete:<number1> Incomplete:<number2>
Ldur:<number3> Hold-> Incomplete:<number4>.
```
The <name> argument represents the name of file that has finished processing, <number1> represents the number of complete calls output, <number2> represents the number of incomplete calls output, <number3> represents the number of long duration calls output, and <number 4> represents the number of incomplete calls. This message indicates that the displayed file has been processed. This is an informational message. No action is required.

```
* COR100 System call failure: <call name> (<error number>)
```
The <call name> represents the name of the failed system call, and <error number> represents the UNIX error number. An error was encountered when this system call was requested. Look up the UNIX error number for more information. Contact the Cisco Technical Assistance Center.

```
(*,**) COR101 File error: <system call> on <filename> (<error number>)
```
The <system call> argument represents the name of the failed system call, <filename> represents the name of the file being operated on, and <error number> represents the UNIX error number. An error was encountered during performance of a system call operation on the specified file. See the UNIX error number for more information. Contact the Cisco Technical Assistance Center.

```
** COR200 Environment variable <variable name> not found.
```
The <variable name> argument represents the name of the environment variable that was not found. This message indicates that the displayed environment variable was not found. Verify that the variable is defined in the sym_defs file, or make sure that it is set before the task is run. Contact the Cisco Technical Assistance Center.

```
(*,**) COR201 Function <function name>() failed, (<error number>)
```
The <function name> argument represents the calling function name, and <error number> represents the UNIX error number. This message indicates the failure of a function call. See the UNIX error number for more information. Contact the Cisco Technical Assistance Center.

```
** COR202 Number of CDR memory extensions exceeded <max number>
```
The <max number> argument represents the maximum number of memory extensions. The maximum number of CDR memory extensions is set too small. Increase the maximum number of CDR memory extensions in the cor_cfg file in the opt/CiscoBAMS/files directory. In that file, the -m value indicates the size of the extents and measures the number of CDRs to save into the memory, and the -e value indicates the number of extents. Contact the Cisco Technical Assistance Center.

```
** COR203 Read incomplete block size <filename>, <filename> expected.
```
The <filename> argument represents the name of the file found and the name of the file expected, respectively. The input file generated by the previous task has an I/O error (either a bad disk block or the disk space is full). The input file is incomplete. COR expected multiple records for each input file. The previous task wrote incomplete records, for example, because of a disk space problem. Contact the Cisco Technical Assistance Center.

```
** COR204 Invalid CDB type: <name>
```
The <name> argument represents the filename found. For more information, check the error file called err_<raw_file_name>. This message indicates that an invalid CDB type was encountered. The COR task writes invalid CDBs to that file. Check the Cisco MGC XECFGPARM.dat file and make sure the MGC CDB type is configured as specified. Contact the Cisco Technical Assistance Center.

```
** COR205 Exhausted all possible sequence numbers: <filename>
```
The <file name> argument represents the name of the file. This message indicates that the number of input files in a single interval exceeded 99,999. Contact the Cisco Technical Assistance Center.

```
** COR206 Multiple boundary crossings in a single file: <filename>
```
The <filename> argument represents the name of the input file. This message is generated if there are more than two boundary crossings in a data file. Contact the Cisco Technical Assistance Center.

* COR207 Errors generating boundary crossing file: <filename>

The <file name> argument represents the name of the file. This message indicates that an error was encountered when the boundary crossing file was generated. Contact the Cisco Technical Assistance Center.

COR208 <string> stat file: <filename>

The <string> argument represents an action, such as "Creating" or "Closing," and <filename> represents the name of the file. This message is generated for the displayed stat file. This is an informational message. No action is necessary.

* COR210 CTL file: <function name> error: <ai string error>

The <function name> argument represents the name of the function, and <ai string error> represents the error code. This message indicates an error returned by a control file function. Contact the Cisco Technical Assistance Center.

* COR211 CTL file measurement value error: <value>

The <value> argument represents the incorrect value. This message indicates that the content of the measurement field in the control file is not 0 or 1. Contact the Cisco Technical Assistance Center.

* COR223 Can't load interval value from check point file

This message indicates an error in the reading of the check point record from the check point file. Delete the check point file, COR_CHECKPT, located in the opt/CiscoBAMS/files directory. Contact the Cisco Technical Assistance Center.

COR225 No check point file to load

This informational message displays when the COR task starts and there is no check point file to load.

COR226 Check point loaded, last interval was <string>, next seq no <number>

The <string> argument represents the current interval string loaded from the check point, and the <number> argument represents the current interval sequence number. This informational message displays when the COR task starts and the check point file is loaded.

* COR227 Interval minutes must be 5, 10, 15, 20 or 30

This message displays when the COR task reads an invalid value from the NODEPARMS configuration file for the interval-minutes. Check the NODEPARMS configuration file using MML and fix the value.

COR228 MAINT_UNBLOCK received for TG without blocks

This informational message displays when a MAINT_UNBLOCK message is received for a trunk group that has no blocked trunks. The CDB is ignored.

COR229 MAINT CDB received with an invalid maint_type

This informational message displays when a MAINT CDB is received for a trunk group without a valid maint_type. The CDB is ignored.

Check Remote Down Task Messages

The Check Remote Down (CRD) task monitors communications between redundant BAMS units, and verifies whether or not the remote unit is down. During processing, this task writes messages to the syslog with a CRD task identifier (for example, CRD001).

* CRD500 Cannot attach to shared memory errno: <error number>

The <error number> argument represents the UNIX error number. The system is unable to access the shared memory segment and displays an error message. Look up the UNIX error for further explanation. Contact the Cisco Technical Assistance Center.

* CRD501 Get Message Queue ID for <message queue name> error, returned <return number>

The <message queue name> represents the name of the message queue, and <return number> represents the number of the returned function. The system is unable to get the message from the sender/receiver message queue and displays the sender/receiver message queue ID. Use the ipcs utility to check the message queue status. Contact the Cisco Technical Assistance Center.

* CRD502 Redundant polling unit is down.
The redundant polling unit is down. Check the remote BAMS unit. If it is down, contact the Cisco Technical Assistance Center.

Note   This alarm is cleared when the alarm condition is removed.

* CRD503 <operation> failed, returned <return number>, errno <error number>
The <operation> argument represents the failed operation, <return number> represents the number of the returned function, and <error number> represents the UNIX error number. The system is unable to send local, send remote, or receive messages and displays an error category and code. Look up the UNIX error. Contact the Cisco Technical Assistance Center.

* CRD504 Receive message time out
The system experienced a receive message timeout. Check the network connection to the remote BAMS unit. Contact the Cisco Technical Assistance Center.

CRD505 Redundant polling unit is up
The redundant polling unit is back on line. This message is displayed for informational purposes. No action is necessary.

CDR to BAF Conversion Task Messages

The CDR to BAF Conversion (CTB) task generates BAF output. During processing, this task writes messages to the syslog with a CTB task identifier (for example, CTB001).

CTB000 <version> <version number> has started, pid <process ID>
The <version> argument represents the task version, the <version number> represents the task version number, and <process ID> represents the UNIX process identifier. The system displays a message that the named task has started. This message is displayed for informational purposes. No action is necessary.

CTB001 <task name> terminating with code <return error code>, pid <process id>
The <task name> argument represents the name of the task, <return error code> represents the returned error code, and <process ID> represents the UNIX process identifier. The named task has terminated abnormally. The code identifies the error encountered by the system. Look up the error code. Contact the Cisco Technical Assistance Center.

CTB002 <task name> terminating with signal <signal number>, <process ID>
The <task name> argument represents the name of the task, <signal number> represents the UNIX signal number, and <process ID> represents the UNIX process identifier. The named task was ended by a UNIX signal. This is an informational message. No action is necessary. See the "Signal Numbers" section on page A-68.

CTB003 <name> terminating normally, pid <number>
The <name> argument represents the task name, and <number> represents the process ID number of the task. This message indicates that the task terminated normally. This is an informational message. No action is necessary.

CTB004 No file to be processed. Retries in <n> seconds
The <n> argument represents the number of seconds during which the task sleeps before accessing more files.

* CTB101 File error <data type> on <system call> (<error number>)
The <data type> argument represents the type of data format, <system call> represents the failed system call, and <error number> represents the UNIX error number. The system encountered an error when performing a system call operation on the specified file. Look up the error number. Contact the Cisco Technical Assistance Center.

```
** CTB200 Environment variable <variable name> not found
```
The <variable name> argument represents the name of the environment variable not found. The system encountered a missing or undefined variable. Either the symbol definitions are missing, or the variable must be set before the task is run. Contact the Cisco Technical Assistance Center.

```
(*,**) CTB201 Function <calling function> failed, (<error number>)
```
The <calling function> argument represents the name of the calling function, and <error number> represents the UNIX error number. The system encountered a failure trying to make a function call. Look up the error number. Contact the Cisco Technical Assistance Center.

```
** CTB202 Invalid AMADEF format, read <error string>, expected <expected string>
```
The <error string> argument represents the error string found, and <expected string> represents the string expected. The system encountered a format problem while trying to read the AMADEF file. The AMADEF file needs to be fixed. Contact the Cisco Technical Assistance Center.

```
* CTB205 Unsupported field <field name>, conversion not defined
```
The <field name> argument represents the name of the field that is not supported. The system encountered an AMA field that cannot be processed. Contact the Cisco Technical Assistance Center.

```
CTB207 Processed <filename>, converted <record number> records, output <output filename>
```
The <filename> argument represents the name of the processed file, <record number> represents the record number, and <output filename> represents the output filename. The system displays the following process information: name of processed file, number of records converted, and output filename. This is an informational message. No action is necessary.

```
(*,**) CTB208 <data type> code <structure code> not found, cannot process record
```
The <data type> argument represents the data type, and <structure code> represents the structure code. The system is unable to process a record due to missing code and structure code information. Check the AMADEF file. A major alarm is generated if an incomplete, on-going call is rejected by the Correlation Task (COR) because an on-going record was not received within 12 hours (based on the cdb_creation_time of the latest data). This record does not have an associated map type. In Release 2.64, this cdr is written out to an error file ebaf* stored in the error directory, /opt/CiscoBAMS/data/error. Contact the Cisco Technical Assistance Center.

```
** CTB209 Error <return value> reading file: CALLSTRUCT_MAP
```
The <return value> represents the error value returned. The system encountered a format error reading the callstruct file. Check the CALLSTRUCT_MAP file. Contact the Cisco Technical Assistance Center.

Message Receive Task Messages

The Message Receive (DRX) task is used by one BAMS unit to receive messages from the other unit in a redundant configuration. During processing, this task writes messages to the syslog with a DRX task identifier (for example, DRX001).

```
DRX000 Message Receive Task version <version number> has started
```
The <version number> argument represents the task version number. The system displays the message at the start of the named task. This message is an informational message. No action is necessary.

```
DRX001 Terminating with signal <signal number>
```
The <signal number> argument represents the UNIX signal number. The system ends the DRX task and displays a signal number. If an operator stops the system, it terminates with signal 15. Look up the . Contact the Cisco Technical Assistance Center.

```
DRX002 Terminating with exit value <signal number>
```
The <signal number> argument represents the UNIX signal number. The system terminates the DRX task and displays a signal number. Look up the . Contact the Cisco Technical Assistance Center.

DRX003 Established connection with remote unit
The system has successfully established a connection with the remote unit. This is an informational message. No action is necessary.

DRX004 Closed connection with remote unit
The system has successfully closed the connection with the remote unit. This is an informational message. No action is necessary.

** DRX005 Cannot wait for connections <error number>
The <error number> argument represents the UNIX error number. The system cannot wait for connection and displays a UNIX error. Look up the error number. Contact the Cisco Technical Assistance Center.

** DRX006 Cannot accept connection <error number>
The <error number> argument represents the UNIX error. The system is unable to accept the connection request and displays a UNIX error. Look up the error number. Contact the Cisco Technical Assistance Center.

** DRX007 Cannot read from connection <error number>
The <error number> argument represents the UNIX error number. The system cannot read from the connection and displays a UNIX error. Look up the error number. Contact the Cisco Technical Assistance Center.

DRX008 Cannot route local message queue; <message queue ID>, <return value>, <error number>
The <message queue ID> argument represents the message queue identifier, <return value> represents the returned value, and <error number> represents the UNIX error number. The system is unable to route the local message to the intended destination and displays a UNIX error. Look up the error number. Use the ipcs utility to check the message queue status. Contact the Cisco Technical Assistance Center.

** DRX009 Connection broken <error number>. Will reconnect
The <error number> argument represents the UNIX error number. The system displays an error message regarding a broken connection. Look up the error number. Check if the remote BAMS unit system was stopped by the operator. Contact the Cisco Technical Assistance Center.

Message Transmit Task Messages

The Message Transmit (DTX) task oversees communication transmissions between BAMS units. During processing, this task writes messages to the syslog with a DTX task identifier (for example, DTX001).

DTX000 Message Transmit Task version <version number> has started
The <version number> argument represents the task version number. The system has started the Message Transmit task. This is an informational message. No action is necessary.

DTX001 Terminating with signal <signal number>
The <signal number> argument represents the UNIX signal. The system is terminating this task and displaying a UNIX signal. Look up the signal number. Check if the system was stopped by the operator if signal number = 15. Contact the Cisco Technical Assistance Center.

DTX002 Terminating with exit value <exit value>
The <exit value> argument represents the UNIX exit value. The system is terminating this task and displaying a UNIX exit value. Contact the Cisco Technical Assistance Center.

DTX003 Established connection with remote unit
The system has established a connection to the remote unit. This is an informational message. No action is necessary.

DTX004 Closed connection with remote unit
The system has closed the connection with the remote unit. This is an informational message. No action is necessary.

DTX005 Could not discard queued messages <returned value> (<error number>)
The <returned value> argument represents the return value, and <error number> represents the UNIX error number. The system is unable to discard queued messages and displays a UNIX error code. Look up the error number. Contact the Cisco Technical Assistance Center.

** DTX006 Connect attempt failed <error number>. Will retry later
The <error number> argument represents the UNIX error number. The system is unable to connect and displays a UNIX error. The system tries to establish the connection again later. Look up the error number. Contact the Cisco Technical Assistance Center.

** DTX007 Failed receiving message <return value> (<error number>)
The <return value> argument represents the returned value, and <error number> represents the UNIX error number. The system is unable to receive a message. Look up the error. Check if the remote BAMS unit is down. Contact the Cisco Technical Assistance Center.

** DTX008 Connection broken <error number>. Will re-connect
The <error number> argument represents the UNIX error number. The system reports an interrupted connection to the remote unit and displays a UNIX error. The system attempts to reconnect to the remote unit. Look up the error number. Check the network connection to the remote BAMS unit. See if the operator stopped the remote BAMS unit or if some other error occurred on that unit. Contact the Cisco Technical Assistance Center.

** DTX009 Cannot write to connection.
The system is unable to transmit information. Contact the Cisco Technical Assistance Center.

*C DTX010 Could not create message queue <return value> (<error number>).
The <return value> argument represents the returned value, and <error number> represents the UNIX error number. The system is unable to create a message queue. Look up the error number. Use the **ipcs** utility to check the message queue status. Contact the Cisco Technical Assistance Center.

* DTX011 Connect socket for host <first host name> failed, trying host <second host name>
The <first host name> argument represents the primary host name for the remote BAMS unit, and <second host name> represents the secondary name. The system was unable to connect to the remote BAMS unit using the first host name and is now trying to connect to the unit using the second host name. Check the host names. Check if a network problem exists that prevents connection to the host represented by <first host name>. Contact the Cisco Technical Assistance Center.

Extendable ASCII Task Messages

The Extendable ASCII (EXT) task generates extendable ASCII data. During processing, this task writes messages to the syslog with an EXT task identifier (for example, EXT001).

EXT000 <task name> version <version> has started pid <process identifier>
The <task name> argument represents the task name, <version> represents the version, and <process identifier> is the UNIX process identifier. This alarm defines what version of the task is running. This is an informational message. No action is required.

EXT001 <task name> terminating with code <return error code>, pid <process id>
The <task name> argument represents the task name, <return error code> represents the error code returned, and <process id> represents the UNIX process identifier. The named task has been terminated abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

EXT002 <task name> terminating with signal <signal number>, pid <process id>
The <task name> argument represents the task name, <signal number> represents the UNIX signal number, and <process id> is the UNIX process identifier. The named task has been ended by a UNIX signal. This is an informational message. No action is necessary.

EXT003 <task name> terminating normally, pid <process id>
The <task name> argument represents the task name, and <process id> is the UNIX process identifier. The EXT task has terminated normally. This is an informational alarm that appears only in test mode.

EXT004 No file to be processed. Retries in <number> seconds
The <number of seconds> argument represents the number of seconds the system waits before reattempting to process files. The task sleeps for the number of seconds displayed before accessing more files. This is an informational message. No action is necessary.

* EXT101 File error: <data format type> on <system call> <errno>
The <data format type> argument represents the type of data format, <system call> represents the name of the failed system call or the name of the file being operated on; and <errno> represents the UNIX error number. An error was encountered during a system call operation. Look up the UNIX error number in the listing at the end of this appendix and contact the Cisco Technical Assistance Center.

** EXT200 Environment variable <name> not found
The <name> argument represents the name of the missing environment variable, for example CONFIG_DIR, FILES_DIR, dir_in, dir_out. The displayed environment variable was not found. The symbol definition file (sym_defs) is either missing a variable, or the variable must be set before the task is run. Contact the Cisco Technical Assistance Center.

(*,**) EXT201 Function <name> failed, <errno>
The <name> argument represents the calling function name, and <errno> represents the UNIX error number. This message indicates failure of the displayed function call, that is, Major Events for open_file, stat (sysParm.CTL), ai_open ctlfile, minor events for ai_get_max_num_of_recs(), ai_load_rec(), or ai_close_file. Examine the error number. Contact the Cisco Technical Assistance Center.

(*,**) EXT201 Function <calling function> failed, (<error number>)
The <calling function> represents the calling function, and the <error number> argument represents the UNIX error number. The system encountered a failure trying to make a function call. Examine the system error. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

** EXT202 Invalid AMADEF format, read <error string>, expected <expected string>
The <error string> argument represents the error string found, and <expected string> represents the string that was expected. The system encountered a format problem while trying to read the AMADEF file. The AMADEF file needs to be fixed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

* EXT205 Unsupported field <field name>, conversion not defined
The <field name> argument represents the name of the field. The system encountered an AMA field that could not be processed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

EXT207 Processed <filename>, converted <num recs>, output = <suffix>
The <filename> argument represents the input filename, <num recs> represents the number of records converted, and <suffix> represents the output file suffix. This message displays processing status information. This is an informational message. No action is necessary.

(*,**) EXT208 <data type> code <structure code> not found, cannot process record
The <data type> argument represents the type of data, and <structure code> represents the structure code or call type. The system is unable to process a record due to missing code and structure code information. Check the AMADEF file. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

** EXT209 Error <return value> reading file: CALLSTRUCT_MAP
The <return value> argument represents the value returned. The system encountered a format error reading the callstruct file. Check the CALLSTRUCT_MAP file. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

Formatter Task Messages

The Formatter (FMT) task formats files. FMT processes the polled raw files from the switch and converts them to an aggregated record format that BAMS uses for all processing tasks. FMT renames the timestamp in the filename to match the timestamp in the raw file header record. During processing, this task writes messages to the syslog with an FMT task identifier (for example, FMT001).

`FMT000 <task> version <version number> has started pid <process ID>.`
The <task> argument represents the task name, <version number> represents the task version number, and <process ID> represents the UNIX process identifier. This message identifies the version of the task that is running. This is an informational message. No action is necessary.

`FMT001 <task> terminating with code <return code>, pid <process ID>.`
The <task> argument represents the task name, <return code> identifies the terminating signal number encountered by the system, and <process ID> represents the UNIX process identifier. The named task ended abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

`FMT002 <task> terminating with signal <number>, pid <process ID>.`
The <task> argument represents the task name, <number> represents the UNIX signal number, and <process ID> represents the UNIX process identifier. The named task was ended by a UNIX signal.

`FMT003 <task> terminating normally, pid <process ID>.`
The <task> argument represents the task name, and <process ID> represents the UNIX process identifier. The named task ended normally. This is an informational message. No action is necessary.

`FMT004 Processing file <filename>`
The <filename> represents the name of the file being processed. The task has started processing the named file. This is an informational message. No action is necessary.

`FMT009 Received system signal <signal>`
The <signal> argument represents the received signal number. The FMT task received a system signal. This is an informational message. No action is necessary.

`FMT010 Processed <filename>, total:<number1> good:<number2> fmterr:<number3>`
`synerr:<number4> semerr:<number5> skipcdb`
The <filename> argument represents the processed filename, <number1> represents the total number of processed call records, <number2> represents the number of good call records, <number3> represents the number of bad format call records, <number4> represents the number of call records with syntax errors, and <number5> represents the number of call records with semantic errors. This message displays call process summary information. This is an informational message. No action is necessary.

`FMT011 Processed <filename>, Skipped CDEs: <number1> Unknown CDEs: <number2>`
The <filename> argument represents the processed filename, <number1> represents the number of skipped CDEs, and <number2> represents the number of unknown CDEs. This message indicates the number of CDEs that were skipped or unknown in a processed file. This is an informational message. No action is necessary.

`FMT012 Output file <filename>`
The <filename> argument represents the processed filename. This message indicates the name of the output file. This is an informational message. No action is necessary.

`FMT019 [CDR] No flat files to process. Retry in <number> seconds.`
The <number> argument represents the number of seconds. This message indicates that no data exists for processing. The FMT task retries in the displayed number of seconds. This is an informational message. No action is necessary.

`FMT020 CDB tag <number>, CDB skipped.`
The <number> argument represents the CDB number. This CDB is skipped when the CDB tag is found in the Skip CDB table. This is an informational message. No action is necessary.

FMT021 CDE tag <number>, CDB skipped.
The <number> argument represents the CDE number. This CDE is skipped when the CDE tag is found in the Skip CDE table. This is an informational message. No action is necessary.

FMT024 Creating <name> table
The <name> argument represents the lookup table name (that is, NPANXX table, Rating Exception table, Rating Type table, or Trunk Group table). The named lookup table is created and loaded into memory. This is an informational message. No action is necessary.

FMT025 Refreshing <name> table.
The <name> argument represents the lookup table name. The lookup table is reloaded. This is an informational message. No action is necessary.

* FMT100 System call failure: <name> (<errno>)
The <name> argument represents the name of a failed system call, and <errno> represents the UNIX error number. This message indicates that the FMT task encountered an error when requesting the displayed system call. Look up the UNIX error number. Contact the Cisco Technical Assistance Center.

* FMT101 File error: <call> on <file> (<code>)
The <call> argument represents the name of the failed system call, <file> represents the filename, and <code> represents the UNIX error number. This message indicates that the FMT task encountered an error when performing the displayed system call operation. Look up the error number. Contact the Cisco Technical Assistance Center.

*C FMT102 File I/O error: <call> on <file>, requested <number1>, got <number2>
The <call> argument represents the name of the failed I/O system call, <file> represents the name of the file being operated on, <number1> represents the number of bytes requested, and <number2> represents the number of bytes received. The FMT task encountered an error when performing the displayed I/O file operation. Contact the Cisco Technical Assistance Center.

* FMT201 Undefined environment variable <name>
The <name> represents the name of the environment variable. This message indicates that the displayed environment variable is not defined. Contact the Cisco Technical Assistance Center.

* FMT202 Input file does not start with header CDB
This message indicates that the FMT task detected an invalid field value in the call record. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* FMT203 Error reading TAG/LEN of TLV errno=<error number>
The <error number> argument represents the UNIX error number. An error occurred during the reading of the TLV (T = Tag field code, L = Length of the field value, V = field Value). The FMT task stops processing this file, renames it as an error file, and begins processing the next input file. Look up the error. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

* FMT204 Input file is not ended by footer CDB
The system encountered the displayed error. Contact the Cisco Technical Assistance Center.

* FMT207 #<index> - <name>: <value> too long, <size> expected.
The <index> argument represents the call record index, the <name> argument represents the field name, the <value> argument represents the field value, and the <size> argument represents the expected field size. This message indicates that the field string exceeded the expected size. Contact the Cisco Technical Assistance Center.

* FMT210 CTL file: <name>:<error>.
The <name> argument represents the control filename, and the <error> argument represents the displayed error string. This message indicates the error returned by a control file function. Contact the Cisco Technical Assistance Center.

* FMT211 CTL file measurement value error: <value>
The <value> argument represents the incorrect value. This message indicates that the content of the measurement field in the control file is not 0 or 1. Contact the Cisco Technical Assistance Center.

Formatter Task Messages

```
* FMT212 CTL file lookup value error: <value>
```
The <value> argument represents the measurement value. This message displays when the value of the measurement field in the control file is not 0 or 1. Check the configuration file using MML and fix the value.

```
(*,**) FMT221 Function <function name>() failed, (<error number>)
```
The <function name> argument represents the calling function name, and <error number> represents the UNIX error number. This message indicates the failure of a function call. See the UNIX error number for more information. Contact the Cisco Technical Assistance Center.

```
* FMT222 Invalid TLV len:<length> at <position>th chars, CDB len:<size>
```
The system read in invalid TLV length. The <length> argument represents the incorrect TLV length, <position> represents the TLV position in the CDB, and <size> represents the size of the CDB. The FMT task stops further processing of the input file and renames the input file by prepending "err_" to the input filename. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

```
* FMT224 Process all CDBs: Error in parsing record.
```
This message indicates that the FMT task encountered an error during the parsing of this record. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

```
* FMT225 Dump_cde: Unrecognized tag <tag number>
```
The <tag number> represents the number of the tag. This message indicates that the FMT task encountered an unrecognized tag. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

```
* FMT226 Incorrect field length <field value length>.
```
The <field value length> argument represents the length of the field value. This message indicates that the FMT task encountered an incorrect field length. Verify the Cisco MGC input file. Contact the Cisco Technical Assistance Center.

```
FMT232 <index> - Fail <function> lookup of <field1>: <value1> and <field2>: <value2>
```
The <index> argument represents the index of the call record, the <function> argument represents the function name of the table lookup, the <field1> argument represents the first field name in the call record, the <value1> argument represents the first value of the field being looked up, the <field2> argument represents the second field name in the call record, and the <value2> argument represents the second value of the field being looked up. This message indicates that the task failed to look up the indicated fields.

```
FMT234 #<record index> - Invalid record <field names/values>
```
The <record index> argument represents the record index, and the <field names/values> argument represents the field names and values. This message is generated if bafoutput (Node Parameters table) is enabled and the FMT task encounters an invalid block format while reading blocks from a control file. The message shows which fields are invalid. Use MML to correct the values.

```
FMT235 <number> record(s) failed <table> lookup for file <name>
```
The <number> argument represents the number of records, the <table> argument represents either the Trunk Group table (TRUNKGRP) or the NAILED Connection table (SIGPATH), and the <name> argument represents the filename. This message shows the number of lookup failures per file. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

Correct the appropriate record in the Trunk Group table (TRUNKGRP) or in the Nailed Connection table (SIGPATH).

Example:

The following example represents an FMT [23470] message from a /opt/CiscoBAMS/files/sxx/FMT_cdr.log file.

```
FMT [23470] #1 - Fail Trunk Group lookup of Orig_trk_grp: 1002
```

The example of an FMT_cdr.log file message includes the following elements:

- [23470]—the process ID of the process on the BAMS node
- #1—the record number
- Orig_trk_grp—the Trunk Group parameter
- 1002—the value that failed lookup

✎

Note   The process ID changes whenever the FMT task restarts.

```
FMT236 Setting FMT processing prefix to <prefix>
```
The <prefix> argument represents the file prefix. The FMT task processes files whose prefixes match the file-prefix1 parameter in the Poll table. If this parameter is not populated, the file-prefix2 parameter is used. If both parameters are empty, the default prefix is cdr. Whenever the Poll table is modified, this message displays the value of the file prefix that FMT uses.

CDR Binary Error Lookup Files

Cisco BAMS generates error lookup (elkup) files when the FMT process encounters lookup failures for a trunk group, NAILED_CONN (SIGPATH), NPA_NXX, RATE_TYPE, or RATE_EXEC, and cannot perform data conversion. The elkup error files are binary CDR files that are saved in the directory /opt/CiscoBAMS/data/s0x/error on the BAMS server.

For example, the file elkup_cdr_20080822155821_006831.bin in the directory /opt/CiscoBAMS/data/s0x/error indicates that some number of lookups failed for cdr_20080822155821_006831.bin. As in the example, for every occurrence, the elkup error file name is formed by the expression elkup_ as a prefix to the CDR bin file name.

If you login as user "bams**,**" you can display a translation of the elkup bin file by issuing the command **cdrdump** *<elkup_cdr_bin_filename>*. The **cdrdump** program is located in the directory /opt/CiscoBAMS/contrib, but can be executed from any directory if the login user is "bams."

The data displayed in the translated bin file includes an error code that identifies the table for which the lookup failed. Table A-15 maps the error code values to the table lookup types.

Table A-15   Elkup File Error Code Identification

Elkup File Error Code	Error Type
0x000100	NPA_NXX
0x000200	TRUNKGROUP
0x000400	RATE_TYPE
0x000800	RATE_EXC
0x001000	NAILED_CONN (SIGPATH)
0x000000	**Note** This code indicates that all executed lookups were successful.

For example, the error code 0x0001200 indicates that the lookup failed for both a NAILED_CONN (SIGPATH) table and TRUNKGROUP table.

The error code value that appears in the elkup file is derived by adding (using HEX arithmetic) the individual error code values of the failed lookup functions.

Consider the derivation of the error codes in the following two examples.

Example 1

1. The error code 0x00001300 results from errors in the NAILED_CONN (0x1000), TRUNKGRP (0x200), and NPA_NXX (0x100) lookup tables.

    0x1000 + 0x200 + 0x100 = 0x00001300

Example 2

2. The error code 0x00000d00 results from errors in both the NPA_NXX (0x100), RATE_TYPE (0x400), and RATE_EXC (0x800) lookup tables.

    0x100 + 0x400 + 0x800 = 0x00000d00

To correct lookup failures, examine the relevant fields in the elkup file and the BAMS tables as described in the following sections.

NPANXX Table Errors

The elkup file error code 0x001000 indicates that a NPANXX table lookup failed. To identify the table entries that are missing or that have failed lookup, check the file FMT_cdr.log in the directory /opt/CiscoBAMS/files/s0x. Specifically, in the FMT_cdr.log file, search for FMT235 messages, which identify the table for which a number of lookup errors occurred. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

If the BAF output has been turned on, there is a lookup failure of the npanxx of either the called_number or calling_number from the elkup record. This npanxx value should be populated in the table NPANXX.

See the section "Formatter Task Messages" in this appendix.

To display the provisioned NPA_NXX table entries, issue the MML command **prov-rtrv:NPANXX:**

To reprovision an NPANXX table entry issue the MML command **prov-add**.

See the section "Updating the NPANXX Table" in Chapter 5.

Trunk Group Table Errors

The elkup file error code 0x000200 indicates that a TRUNKGROUP table lookup failed. To identify the table entries that are missing or that have failed lookup, check the file FMT_cdr.log in the directory /opt/CiscoBAMS/files/s0x. Specifically, in the FMT_cdr.log file, search for FMT235 messages, which identify the table for which a number of lookup errors occurred. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

The following sample display was extracted from a FMT_cdr.log file. The data confirms a specific trunk group lookup failure.

```
09/05/08 00:09:47 FMT [ 6890] #9525 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9526 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9527 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9528 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9529 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9530 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9531 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9532 - Fail Trunk Group lookup of Orig_trk_grp: 7000
09/05/08 00:09:47 FMT [ 6890] #9533 - Fail Trunk Group lookup of Orig_trk_grp: 7000
```

For trunk group errors, you should examine the values of orig_trk_grp, term_trk_grp, and maint_trk_grp in the elkup record. If any of these are populated, a corresponding entry should be present in the table TRUNKGRP. A lookup error indicates that at least one of the fields (orig, term, or maint) failed the TRUNKGRP lookup.

See the section "Formatter Task Messages" in this appendix.

To display the provisioned TRUNKGRP table entries, issue the MML command **prov-rtrv:TRUNKGRP:**

To reprovision a TRUNKGRP table entry issue the MML command **prov-add**.

See the section "Updating the Trunk Group Table" in Chapter 5.

Rating Type Table Errors

The elkup file error code 0x000400 indicates that a RATING-TYPE table lookup failed. To identify the table entries that are missing or that have failed lookup, check the file FMT_cdr.log in the directory /opt/CiscoBAMS/files/s0x. Specifically, in the FMT_cdr.log file, search for FMT235 messages, which identify the table for which a number of lookup errors occurred. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

If the BAF output is turned on, the field call_rate_type is populated from either the table RATE_TYPE. If the error was RATE_TYPE_LOOKUP, then lookup of the fields igr_call_zone_id or egr_call_zone_id from the elkup record failed in the table RATE_TYPE.

See <Cross reference the section "Formatter Task Messages" in this appendix.

To display the provisioned RATING-TYPE table entries, issue the MML command **prov-rtrv:RATING-TYPE:**

To reprovision a RATING-TYPE table entry issue the MML command **prov-add**.

See the section "Updating the Rating Type Table" in Chapter 5.

Rating Exception Table Errors

The elkup file error code 0x000800 indicates that a RATE-EXC table lookup failed. To identify the table entries that are missing or that have failed lookup, check the file FMT_cdr.log in the directory /opt/CiscoBAMS/files/s0x. Specifically, in the FMT_cdr.log file, search for FMT235 messages, which identify the table for which a number of lookup errors occurred. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

In this case, if the BAF output has been turned on, the field call_rate_type is populated from either the table RATE_EXC. If the error was RATE_EXC_LOOKUP, then the lookup of the npanxx of either the called_number or calling_number from the elkup record failed in the table RATE_EXC.

See the section "Formatter Task Messages" in this appendix.

To display the provisioned RATE-EXC table entries, issue the MML command **prov-rtrv:RATE-EXC:**

To reprovision a RATE-EXC table entry issue the MML command **prov-add**.

See the section "Updating the Rating Exception Table" in Chapter 5.

Nailed Connection Table Errors

The elkup file error code 0x000800 indicates that a SIGPATH table lookup failed. To identify the table entries that are missing or that have failed lookup, check the file FMT_cdr.log in the directory /opt/CiscoBAMS/files/s0x. Specifically, in the FMT_cdr.log file, search for FMT235 messages, which identify the table for which a number of lookup errors occurred. To find details of a lookup failure, examine the log file /opt/CiscoBAMS/files/sxx/FMT_cdr.log and search for FMT [23470] messages.

If the BAMS system is in the nailed configuration mode, you should examine the values of igr_sigpath_id, egr_sigpath_id, and maint_sigpath_id in the elkup record. If any of these are populated, a corresponding entry should be present in the table NAILED_CONN. A lookup error indicates that at least one of the fields (igr, egr, or maint) failed the NAILED_CONN lookup.

See the section "Formatter Task Messages" in this appendix.

To display the provisioned SIGPATH table entries, issue the MML command **prov-rtrv:SIGPATH:**

To reprovision a SIGPATH table entry issue the MML command **prov-add: sigpath**.

See the section "Updating the Nailed Connection Table" in Chapter 5.

Man-Machine Language Messages

The Man-Machine Language (MML) task supports the command-line interface. This task also displays syntax error messages if the operator makes a mistake entering a command. The syntax error messages in the following paragraphs are generally not written to the syslog file.

`* MML600 Missed ';' in MML_REC_MAP file, line number <#>`
The <#> argument indicates which line has a problem. The system encountered a syntax error in the record map file found in the opt/CiscoBAMS/files directory. Each line in the record map file has a maximum of 3 semicolon-separated fields. Use vi to look at the file and insert the semicolon field separator, if it is missing. Contact the Cisco Technical Assistance Center

`* MML601 Open control file <name> failed`
The <name> argument represents the name of the control file that the MML task could not open. The system was unable to open the displayed control file. Verify that the specified control file exists in the opt/CiscoBAMS/files directory, and that the group permissions (775) and ownership (bams) are properly set up. Contact the Cisco Technical Assistance Center.

`* MML603 Load record <number> failed`
The <number> argument represents the record number. The system was unable to load the displayed record. Contact the Cisco Technical Assistance Center.

`* MML604 Set field <name> failed for record <number>`
The <name> argument represents the field name that the MML task could not set, and <number> represents the call record number. The system was unable to set a field name for the displayed record. Contact the Cisco Technical Assistance Center.

`* MML605 Update record <number> failed`
The <number> argument represents the number of the record that the MML task was unable to update. The system was unable to update the displayed record. Contact the Cisco Technical Assistance Center.

`* MML606 Close control file <name> failed`
The <name> argument represents the name of the control file that the MML task was unable to close. The system was unable to close the displayed control file. Contact the Cisco Technical Assistance Center.

`(*,**,*C) MML608 Get number of keys failed`
The system was unable to get the number of keys. Contact the Cisco Technical Assistance Center.

(*,**,*C) MML609 Missing row number or key field
The system encountered a missing row number or key field. Contact the Cisco Technical Assistance Center.

(*,**,*C) MML610 Get number of fields failed
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

(*,**,*C) MML611 Get field names failed
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

* MML612 Get field names error, NULL field name encountered
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

* MML613 Get fields <name> failed for record <number>
The <name> argument represents the name of the field that the MML task was unable to get, and <number> represents the record number. The system was unable to get the displayed fields for the displayed record. Contact the Cisco Technical Assistance Center.

** MML614 Error finding type for field <name>. May have problem with CFDT file.
The <name> argument represents the field name. This message indicates an error encountered in finding the displayed field name type. There might be a problem in the configuration file. Contact the Cisco Technical Assistance Center.

(*,**,*C) MML615 Get key names failed
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

(*,**.*C) MML616 Get max number of records failed
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

(*,**,*C) MML617 Max number of records should be greater than 0
The system experienced the displayed failure. Contact the Cisco Technical Assistance Center.

*C MML618 Dir <destination directory> could not be created.
The <destination directory> variable represents the destination directory name. The system was unable to create a destination directory. Contact the Cisco Technical Assistance Center.

* MML619 System call <name> failed. errno=<error number>
The <name> argument represents the name of the failed system call, and <error number> represents the UNIX error number. A system call failed. Look up the error number. Contact the Cisco Technical Assistance Center.

MML621 Successful prov-dply from <source directory>.
The <source directory> argument represents the file location specified with the **prov-dply** command. This is an informational message. No action is necessary.

(*, **, *C) MML622 File <number> could not be opened for <filename>.
The <number> argument represents the record number, and <filename> represents the control filename. The file might not exist or might have become corrupt. Check the file. Contact the Cisco Technical Assistance Center for further assistance.

MML623 Read record <number> failed for control <file>.
The <number> argument represents the record number, and <filename> represents the control filename. The control file might be in use. Try again in a few minutes. Contact the Cisco Technical Assistance Center for further assistance.

Mass Storage Control Task Messages

The Mass Storage Control (MSC) task monitors and conserves disk space on the BAMS units. It generates alarms when disk occupancy passes user-defined thresholds based on file prefixes found in specified data directories, and deletes files to free disk space.

Polled files are files that have been polled by a polling computer. Unpolled files are files that have not been polled yet. The MSC task can be set up to delete polled files when the percentage of disk space occupied by these files exceeds the user-defined thresholds. For example, if the polled upper threshold is set at 80 percent of disk occupancy and the lower polled threshold is set at 60 percent, the MSC task begins deleting polled files from oldest to newest when the total polled file size exceeds the 80 percent upper threshold. It stops deleting these files when it falls below the lower threshold. The unpolled threshold value is usually set high, for example, 90 percent, to avoid losing unpolled data. An alarm is generated to alert the user that the disk is filling up.

During processing, this task writes messages to the syslog with an MSC task identifier (for example, MSC001).

MSC001 MSC version <version> cleaning disk
The <version> argument represents the version number of the MSC task. The MSC task periodically deletes user-specified files from the disk at user-defined intervals so that the disk does not fill up and cause system failure. This message indicates that the MSC task is currently working. This is an informational message. No action is necessary.

* MSC003 <string1> error <UNIX error number> on <string2>
The <string1> argument represents the path or full filename and statistics. The <UNIX error number> argument represents the UNIX error set by the system function. The <string2> argument represents the path, or full filename, and statistics. This message indicates that an error occurred while the MSC task was searching for files to set up for deletion. An informational message is recorded in the system log file if the MSC task cannot obtain information on the file. A minor event is recorded if an error occurs while the MSC task attempts to retrieve information on the number of inodes, free disk space, and other similar information.

Check the UNIX error number to determine the reason for the error. Log in to BAMS and source the sym_defs file. Use the **make nodes** and **make home** commands, restart BAMS, and monitor the system log file to see if the error persists. If it does, restart BAMS to clear system resources.

* MSC006 Error <UNIX error number> opening <file>
The <UNIX error number> argument represents the UNIX error set by the system function. The <file> argument represents the filename. This message indicates that an error occurred while BAMS was attempting to locate a file within the subdirectories of the /CiscoBAMS/data directory.

Check the UNIX error number to determine the reason for the error. Verify that the msct.parm file exists and that the permissions are correct. Contact the Cisco Technical Assistance Center.

* MSC007 Invalid parameter file
This message indicates that the parameter file for the MSC task has a problem. The file might be corrupted or it might have an invalid parameter that was saved during MSC maintenance.

Verify that the msct.parm file exists and that the permissions are correct. Verify the information in the file. Contact the Cisco Technical Assistance Center.

MSC008 Maximum number of files exceeded
This message indicates that the MSC task attempted to clean up the disk by deleting files specified by the data storage attributes and has found that there are too many files on the disk. This is an informational message. No action is necessary.

Log in to BAMS and check disk usage. Delete data files from BAMS manually. The data storage attributes might not be set correctly; check them to ensure that this situation does not recur.

```
MSC009 Error <UNIX error number> opening DISK_STATS file
```
The <UNIX error number> argument represents the UNIX error set by the system function. This message indicates that BAMS tried to open the DISK_STATS file and encountered the specified UNIX error. The DISK_STATS file shows the amount of disk space currently being used on BAMS.

Log in to BAMS and use the **ls** command (**ls DISK_STATS**) to check if the DISK_STATS file is on the disk.

```
* MSC010 Undefined environment variable: <variable>
```
The <variable> argument represents the undefined variable. This message indicates that the MSC task encountered an error while attempting to run the housekeeping utility for deleting older data files. The data storage utility verifies that the path for the data directory is defined and can get the specific attributes from the msct.parm file.

Verify that the /CiscoBAMS/data directory is defined in the sym_defs file (located in the /CiscoBAMS/bin directory). Verify that the filenames set in the deletion list are correct (they are case-sensitive).

```
* MSC012 Cannot convert regular expression <reg>
```
The <reg> argument represents the regular expression. This message indicates that BAMS could not recognize the displayed regular expression attribute. Verify the expression used on the MSC menu through the user interface. Contact the Cisco Technical Assistance Center.

```
* MSC014 regBAMS() returned NULL for pattern <reg>
```
The <reg> argument represents the regular expression. This message indicates that BAMS provided a null value in place of the displayed regular expression. Verify the expression used on the MSC menu through the user interface.

```
(*,**,*C) MSC015 <file> is <number> days old, exceeded <max number> day limit
```
The <file> argument represents the name of the file. The <number> argument represents the number of days that the file has resided on the system, and the <max number> argument represents the user-defined file aging limit. The file aging limit (0 to 99 days) is set with the agealarm parameter in MSC-PARMS. This message indicates that BAMS has identified the displayed file as being too old to remain on the system. Contact the Cisco Technical Assistance Center.

```
(*,**,*C) MSC016 <file> is <number> days old, exceeded <max number> day limit DELETED.
```
The <file> argument represents the name of the file. The <number> argument represents the number of days that the file has resided on the system and the <max number> argument represents the user-defined deletion age. The file deletion age (0 to 99 days) is set with the delage parameter in MSC-PARMS. This message indicates that BAMS has identified the displayed file as being too old to remain on the system and has deleted the file. Contact the Cisco Technical Assistance Center.

```
(*,**,*C) MSC017 Disk block utilization <number> percent.
```
The <number> argument represents the number of blocks as a user-defined percentage of the disk block utilization. This minor, major, or critical alarm level is set with the mil, mal, or crl parameter in the MSC Thresholds table. This message identifies the number of blocks used in the file system. The alarm level depends on the percentage of disk block usage. No action is necessary, unless there is a very high percentage of disk block usage. In that case, contact the Cisco Technical Assistance Center.

```
(*,**,*C) MSC018 Disk inode utilization <number> percent
```
The <number> argument represents the number of inodes as a percentage of the total number of inodes. This inode upper threshold is set with the iut parameter in the MSC Thresholds table. This message identifies the number of blocks used in the file system. The alarm level depends on the percentage of inode usage. No action is necessary, unless there is a very high percentage of inode usage. In that case, contact the Cisco Technical Assistance Center.

```
MSC019 MSC waiting <number> minutes before starting.
```
The <number> argument represents the number of minutes the task will wait before starting. This is set with the interval parameter in the MSC Thresholds table. The task waits the displayed number of minutes before starting. This is an informational message. No action is necessary.

```
* MSC020 Access error trying to access <directory path>
```
The <directory path> argument represents the directory path. Search permission has been denied on a component of the path prefix. Verify the directory path and try again. Contact the Cisco Technical Assistance Center.

```
* MSC021 Not a valid directory <directory path>
```
The <directory path> argument represents the directory path that is not valid. A component of the path prefix of <directory path> is not a directory. Verify the directory path and try again. Contact the Cisco Technical Assistance Center.

```
* MSC022 The directory does not exist <directory path>
```
The <directory path> argument represents the missing directory or file. Either a component of the path prefix or the file referred to by path does not exist. Verify the directory path and try again. Contact the Cisco Technical Assistance Center.

```
*C MSC030 Data Disk block utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk utilization. This critical alarm level is set with the crl parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
** MSC031 Data Disk block utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk utilization. This major alarm level is set by the mal parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
* MSC032 Data Disk block utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk utilization. This minor alarm level is set with the mil parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
*C MSC040 Data Disk inode utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk inode utilization. This critical alarm level is set with the crl parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
** MSC041 Data Disk inode utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk inode utilization. This major alarm level is set with the mal parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

> **Note**   This alarm is cleared after the error alarm condition is removed.

```
* MSC042 Data Disk inode utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk inode utilization. This minor alarm level is set with the mil parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

Note   This alarm is cleared after the error alarm condition is removed.

```
*, **, *C MSC044 <filename> is <number of days> days old, exceeded <number> day limit,
DELETED
```
The <filename> argument represents the full name of the file, <number of days> represents the number of days the file has aged, and <number> represents the aging limit in days. This message indicates that the displayed file age has exceeded the aging limit, as set in the MSC Thresholds table, and the file has been deleted.

```
*, **, C MSC045 <filename> is <number of days> old, exceeded <number> day warning age
```
The <filename> argument represents the full name of the file, <number of days> represents the number of days the file has aged, and <number> represents the aging limit in days. This message indicates that the displayed file age has exceeded the warning limit defined in the MSC Thresholds table.

```
*, **, *C MSC046 Disk block utilization: <number> percent, <filename> DELETED
```
The <filename> argument represents the full name of the file, and <number> represents the disk block utilization as a percentage of the total available disk space. This message indicates that the displayed file was deleted because the disk usage exceeded the block threshold defined in the MSC Thresholds table.

```
*,**,*C MSC047 Disk inode utilization: <number> percent, <filename> DELETED
```
The <filename> argument represents the full name of the file, and <number> represents the inode utilization as a percentage of the total available inodes. This message indicates that the displayed file was deleted because the inode usage exceeded the inode threshold, as set with the iut parameter in the MSC Thresholds table.

```
*,**,*C MSC048 Total inodes: <number> exceeded <maxnumber> inode limit, <filename> DELETED
```
The <filename> argument represents the full name of the file, and <number> represents the inode utilization as a percentage of the total available inodes, <maxnumber>. This message indicates that the displayed file was deleted because the inode usage exceeded the inode limit, as set with the iut parameter in the MSC Thresholds table.

```
*, **, *C MSC049 Disk block utilization: <number> percent in partition, <partition>
```
The <partition> argument represents the partition name, and <number> represents the disk block utilization as a percentage of the available partition disk space. This message indicates that the partition thresholds, as defined in the MSC Thresholds table, were exceeded.

```
MSC100 Update MSC task parameters
```
The parameters in the MSC configuration file have been updated and saved to disk. Someone has logged on as an administrative user and has modified the MSC parameters. Administrative users have access to all of the operator-level features, and they can modify system parameters and disk files. This is an informational message. No action is necessary.

```
* MSC101 Error <UNIX error number> running script: <name>
```
The <UNIX error number> argument represents the UNIX error set by the system function. The <name> argument represents the script name. This message indicates that the MSC task experienced an error while running the displayed script from the command line. Look up the error number. Contact the Cisco Technical Assistance Center.

```
*C MSC130 Archive Disk block utilization <number> percent
```
The <number> argument represents the threshold value as a percentage of disk utilization. This critical alarm level is set with the crl parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

** MSC131 Dir Entry 1: Archive Disk block utilization <number> percent
The <number> argument represents the threshold value as a percentage of disk utilization. "Dir Entry 1:" refers to the archive disk partition. This major alarm level is set with the mal parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

* MSC132 Dir Entry 1: Archive Disk block utilization <number> percent
The <number> argument represents the threshold value as a percentage of disk utilization. "Dir Entry 1:" refers to the archive disk partition. This minor alarm level is set with the mil parameter in the MSC Thresholds table. This message reports on disk usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

*C MSC140 Dir Entry 1: Archive Disk inode utilization <number> percent
The <number> argument represents the threshold value as a percentage of disk inode utilization. "Dir Entry 1:" refers to the archive disk partition. This critical alarm level is set with the crl parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

** MSC141 Dir Entry 1: Archive Disk inode utilization <number> percent
The <number> argument represents the threshold value as a percentage of disk inode utilization. "Dir Entry 1:" refers to the archive disk partition. This major alarm level is set with the mal parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

* MSC142 Dir Entry 1: Archive Disk inode utilization <number> percent
The <number> argument represents the threshold value as a percentage of disk inode utilization. "Dir Entry 1:" refers to the archive disk partition. This minor alarm level is set with the mil parameter in the MSC Thresholds table. This message reports on disk inode usage when thresholds are exceeded. Contact the Cisco Technical Assistance Center.

✎
Note   This alarm is cleared after the error alarm condition is removed.

NIC Conversion Task Messages

The NIC task outputs NICS format billing records. During processing, this task writes messages to the syslog with a NIC task identifier (for example, NIC001).

`NIC000 <task name> version <version> has started pid <process identifier>`
The <task name> argument represents the task name, <version> represents the version, and <process identifier> is the UNIX process identifier. This alarm defines what version of the task is running. This is an informational message. No action is required.

`NIC001 <task name> terminating with code <return error code>, pid <process id>`
The <task name> argument represents the task name, <return error code> represents the error code returned, and <process id> represents the UNIX process identifier. The named task has been terminated abnormally. The code identifies the error encountered by the system. This is an informational message. No action is necessary.

`NIC002 <task name> terminating with signal <signal number>, pid <process id>`
The <task name> argument represents the task name, <signal number> represents the UNIX signal number, and <process id> is the UNIX process identifier. The named task has been ended by a UNIX signal. This is an informational message. No action is necessary.

`NIC003 <task name> terminating normally, pid <process id>`
The <task name> argument represents the task name, and <process id> is the UNIX process identifier. The NIC task has terminated normally. This is an informational alarm that appears only in test mode.

`NIC004 No file to be processed. Retries in <number> seconds`
The <number> argument represents the number of seconds the system waits before reattempting to process files. The task sleeps for the number of seconds displayed before accessing more files. This is an informational message. No action is necessary.

`* NIC101 File error: <data format type> on <system call> <errno>`
The <data format type> represents the type of data format; <system call> represents the name of the failed system call or the name of the file being operated on; and <errno> represents the UNIX error number. An error was encountered during the performance of a system call operation. Look up the UNIX error number in the listing at the end of this appendix and contact the Cisco Technical Assistance Center.

`* NIC102 Cannot attach to shared memory, errno <errno>`
The <errno> argument represents the UNIX error number. The attach-to-shared-memory operation failed. Look up the UNIX error number in the listing at the end of this appendix and contact the Cisco Technical Assistance Center.

`** NIC200 Environment variable <name> not found`
The <name> argument represents the name of the missing environment variable, for example CONFIG_DIR, FILES_DIR, dir_in, or dir_out. The displayed environment variable was not found. The symbol definition file (sym_defs) is missing a variable, or the variable must be set before the task is run. Contact the Cisco Technical Assistance Center.

`(*,**) NIC201 Function <name> failed, <errno>`
The <name> argument represents the calling function name, and <errno> represents the UNIX error number. This message indicates failure of the displayed function call, that is, Major Events for open_file, stat (sysParm.CTL), ai_open ctlfile, minor events for ai_get_max_num_of_recs(), ai_load_rec(), or ai_close_file. Examine the error number. Contact the Cisco Technical Assistance Center.

`(*,**) NIC201 Function <calling function> failed, (<error number>)`
The <calling function> represents the calling function, and the <error number> argument represents the UNIX error number. The system encountered a failure trying to make a function call. Examine the system error. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

** NIC202 Invalid AMADEF format, read <error string>, expected <expected string>
The <error string> argument represents the error string found, and the <expected string> argument represents the string that was expected. The system encountered a format problem while trying to read the AMADEF file. The AMADEF file needs to be fixed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

* NIC205 Unsupported field <field name>, conversion not defined
The <field name> argument represents the name of the field. The system encountered an AMA field that cannot be processed. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

NIC207 Processed <filename>, converted <num recs>, output = <suffix>
The <filename> argument represents the input filename, <num recs> represents the number of records converted, and <suffix> represents the output file suffix. This message displays processing status information. This is an informational message. No action is necessary.

(*,**) NIC208 <data type> code <structure code> not found, cannot process record
The <data type> argument represents the type of data and <structure code> represents the structure code or call type. The system is unable to process a record due to missing code and structure code information. Check the AMADEF file. Contact the Cisco Technical Assistance Center for help in checking the system configuration and making any necessary modifications.

* NIC209 Switch name not available in data, set switch name to NULL
The switch name is not available in the data records. The switch name is used to create the NIC output file, so this field is missing (set to NULL) in the file name. Check the data configuration on the switch.

* NIC210 Remote RSN: <seqno1>, local RSN: <seqno2>, setting local value to remote value
The <seqno1> argument represents the remote NICS record sequence number (RSN), and <seqno2> represents the local NICS record sequence number. The RSN value of the remote BAMS unit was greater than the local unit. The system is setting the local value to remote. There is a chance that the RSN values for the NICS files from the two BAMS units do not match. If there has been a rotation of the polling of the NICS files from the BAMS units in the past day, there may be a gap or duplication of the RSN values of the NICS records.

* NIC211 Remote RSN file <name> not received from remote BAMS
The <name> argument represents the filename of the remote RSN file. The remote RSN file was not received from the remote BAMS unit. The NIC task will wait 1 hour for this file. The RSN synchronization will be performed again at midnight (GMT) of the next day. There is a chance that the RSN values for the NICS files from the two BAMS units do not match. If there has been a rotation of the polling of the NICS files from the BAMS units in the past day, there may be a gap or duplication of the RSN values of the NICS records.

* NIC212 RSN sync files are different. Cannot sync RSN values. Local:<name1>, remote:<name2>
The <name1> argument represents the filename of the first file of the day of the local machine, and <name2> represents the filename of the first file of the day of the remote machine. The NIC task will synchronize the NICS record sequence numbers for the first file of each day. If these files are not the same, the RSN synchronization will be aborted. The RSN synchronization will be performed again at midnight (GMT) of the next day. There is a chance that the RSN values for the NICS files from the two BAMS units do not match. If there has been a rotation of the polling of the NICS files from the BAMS units in the past day, there may be a gap or duplication of the RSN values of the NICS records.

NIC213 File <name> not found. Resetting RSN to 1
The <name> arguments represents the name of the file (that is, a NICS RSN sequence number file - NICS_RSN). The RSN seqno file was not found. The system reset the RSN to 1.

NIC214 NIC program waiting up to 1 hour for record sequence number (RSN) sync file from remote BAMS: <path>
The <path> argument represents the pathname of the remote RSN file. The NIC task is looking for the remote RSN file to synchronize the RSN values.

```
NIC215 RSN sync aborted. Remote BAMS is detected down
```
The RSN synchronization will be attempted again at midnight (GMT) of the next day. Check the remote BAMS unit to see why it is detected down.

Node Name Link Task Messages

The Node Name Link (NNL) task synchronizes MML commands to the remote BAMS unit. During processing, this task writes messages to the syslog with an NNL task identifier (for example, NNL001).

```
NNL101 NNL version 1.01 has started pid <pid>
```
The <pid> argument represents the process identifier. The NNL task has been started. This is an informational message. No action is necessary.

```
NNL102 NNL terminating with signal <signal number>
```
The <signal number> argument represents the UNIX signal number. The NNL task is stopping. This is an informational message. No action is necessary.

```
NNL500 Node <number>: changing nodename <oldname> to <newname>
```
The <number> argument represents the node number. The <oldname> and <newname> arguments represent the old node name and the new node name, respectively. The node name has been modified through the MML command **set-nodename**. The node name on the local unit has been changed by the MML command. The node name on the remote unit is synchronized by the NNL task. This is an informational message. No action is necessary.

```
NNL514 Generate alarm <alarm ID> for node <node number>. Level = <alarm level>
```
An alarm was generated from the user interface (mml) of the remote BAMS units. The NNL task generates the same alarm on the local BAMS unit.

Redundant Unit Polling Task Messages

The Redundant Unit Polling (POL) task collects data from the Cisco MGCs. During processing, this task writes messages to the syslog with a POL task identifier (for example, POL001).

```
POL000 Entering <name>()
```
The <name> argument represents the function name. The named function has been entered. This is an informational message. No action is necessary.

```
POL001 Leaving <name>()
```
The <name> argument represents the function name. The named function has been left. This is an informational message. No action is necessary.

```
POL002 Assertion failed in <name>, line <number>
```
The <name> argument represents the function name and the <number> argument represents the line number. An assertion failed in the named function. This is an informational message. No action is necessary.

```
POL100 POL Version <version number> for site <site number> has started
```
The <version number> argument represents the software or task version, and <site number> represents the polling site number. The POL task has been started. This is an informational message. No action is necessary.

```
POL104 Polling VSC unit <MGC node name>
```
The <MGC node name> argument represents the name of the Cisco MGC. The POL task has started to poll the named Cisco MGC unit. This is an informational message. No action is necessary.

*POL105 Polling unit rotates from unit <from unit number> to <to unit number> successfully
The <from unit number> argument represents the number of the unit that was actively polling before the polling unit rotation and the <to unit number> argument represents the number of the unit that was the standby unit before the polling unit rotation. The unit responsible for polling has rotated as displayed.

POL107 File <filename> received from unit <from unit number> successfully
The <filename> argument represents the name of the file that was successfully received from the polling unit and the <from unit number> represents the remote polling unit number. The POL task has received a file as displayed. This is an informational message. No action is necessary.

POL108 Retry #<retry times> will start in <number>
The <retry times> argument represents the number of times the task has retried to poll the site, and the <number> argument represents the number of seconds the task waits before attempting to poll the site again. The POL task retries to poll the data. This is an informational message. No action is necessary.

POL109 Successfully polled <filename> with size <file size>
The <filename> argument represents the name of the file that was successfully polled and the <file size> argument represents the size of the polled file. The POL task has polled the named file. This is an informational message. No action is necessary.

POL110 Cannot update CTLFILE from polling unit to standby unit
The task performed an illegal operation in attempting to update the control file. Check the task permissions and times. The control file is locked if either unit is actively polling, and it can be modified only when both units are at rest.

POL111 This is standby unit
The identified unit is not the current polling unit. This is an informational message. No action is necessary.

POL112 Polling VSC host <host name1> failed, trying host <host name2>
The <host name1> argument represents the name of the first Cisco MGC host line polled and the <host name2> argument represents the name of the Cisco MGC host for the second Ethernet card. The system's attempt to poll the first Cisco MGC host failed, and it is now trying to poll the Cisco MGC host for the second Ethernet card. This is an informational message. No action is necessary.

* POL113 Node is inactive, session terminated
Polling has been disabled. This minor message occurs once when a node is provisioned from the active state to the inactive state with the activate parameter in the Node Parameters table. Once this alarm is set, it is not cleared until the node is once again activated or the user clears it. Investigate to make sure that node de-activation was intentional because switch files will no longer be polled for this node. Clear the alarm through the mml command **clr-alm**. If polling should not have been deactivated, then set the activate parameter in the Node Parameters table to ENABLE (1), and the alarm will be cleared automatically.

POL114 Rename operation failed for file <filename> on polled unit <unit number>
The <filename> argument represents the file that could not be renamed and the <unit number> argument represents the polled unit number. After a successful poll of a file from the VSC, BAMS was not able to rename the file with a *.finished suffix. Check the polling directory on the VSC units to make sure the login user for the VSC has sufficient permissions to delete and rename files within this directory. The POL task does not reattempt to rename or delete files after they have already been successfully polled, so you must manually rename or delete these files on the VSC machines.

** POL115 Rename operation failed on polled unit <unit number>, this alarm must be
manually cleared
The <unit number> argument represents the polled unit number. This major alarm is generated the first
time that a file rename operation fails. It is not generated again until this alarm has been manually cleared
by the operator and the rename operation fails again. Check the polling directory on the VSC units to
make sure the login user for the VSC has sufficient permissions to delete and rename files within this
directory. The POL task does not reattempt to rename or delete files after they have already been
successfully polled, so you must manually rename or delete these files on the VSC machines. This alarm
must be manually cleared through use of the MML command **clr-alm**.

POL116 Search unit <unit number> for files matching <file pattern>
The <unit number> argument represents the BAMS unit number, and the <file pattern> argument
represents the polled file pattern. On the *active* polling unit, this message is displayed for each polling
session. This message indicates the unit number of the polled unit. Also shown is the file pattern of the
polled file, consisting of the file-prefix, start sequence number, and file-suffix parameters. No action is
necessary.

POL201 POL session ended normally
The poll session ended. This is an informational message. No action is necessary.

POL202 POL session from BAMS unit <unit number> starts
The <unit number> argument represents the BAMS unit number. The poll session has started. This is an
informational message. No action is necessary.

* POL203 chgno executed, unit: <unitno> old: <seqno> new: <seqno>
The <unit no> represents the unit number of the BAMS unit, <seqno> represents the old sequence
number, and the second <seqno> represents the new sequence number. The sequence number has
changed. This is recorded as a minor event for audit purposes only.

* POL204 chgno cannot execute, POL record is in use
The sequence number cannot be changed, because the control file is in use. Try again later.

* POL300 Environment variable <environment variable name> is undefined
The <environment variable name> argument represents the name of the environment variable. The
named environment variable is not defined. Contact the Cisco Technical Assistance Center.

* POL302 regcmp returned NULL for pattern <pattern>
The <pattern> argument represents the polling pattern. This message indicates that an error occurred in
a regular expression comparison and that the file pattern does not match. Verify the polling parameters
and file prefix and suffix values. Contact the Cisco Technical Assistance Center.

* POL305 Invalid operation: <operation number>
The <operation number> argument represents the invalid operation. The system was unable to perform
the specified FTP or SFTP operation; it is not valid. Contact the Cisco Technical Assistance Center.

* POL306 Cannot open <filename>
The <filename> argument represents the name of the file that cannot be opened. The system is unable to
open the named file. This message indicates that the polling task cannot open the named file on the
BAMS unit. Check the directory permissions and file ownership. Contact the Cisco Technical Assistance
Center.

* POL307 Poll failed for <filename>
The <filename> argument represents the name of the file that could not be polled. The system was unable
to poll the named file. Manually test whether FTP is working. Create a test file on the Cisco MGC and
manually FTP or SFTP it to the BAMS unit. This message might indicate a Cisco MGC or network
problem. Contact the Cisco Technical Assistance Center.

* POL308 Cannot get file status of <filename>
The <filename> argument represents the name of the file. The system was unable to obtain a file status
for the named file. Check if the named file exists on the local BAMS unit. Check the file, directory, and
ownership permissions. Contact the Cisco Technical Assistance Center.

* POL309 File size incorrect for polling <filename>

The <filename> argument represents the name of the file to be polled. The polling file size is not correct. The polling task compares the size of the named file with what is expected from the Cisco MGC. Check whether a test file can be transferred by FTP or SFTP to the BAMS unit from the Cisco MGC. If the file size is different, a network problem is indicated. Contact the Cisco Technical Assistance Center.

* POL312 Failed updating control record file

The POL task was unable to update the control record file. This message indicates that the control file is locked. Use the **who** command to check if another user is logged in and might have opened the control file. Check the MML command history in the MML daily log file. Contact the Cisco Technical Assistance Center.

* POL321 Cannot open directory file <directory name>

The <directory name> represents the name of the directory. The POL task cannot open the directory. If the problem persists, contact the Cisco Technical Assistance Center.

* POL322 Parent process: system() call \"<command name>"\ failed, errno=<error number>

The <command name> argument represents the name of the command on which the system call failed, and the <error number> argument represents the UNIX error number. The parent POL task displays an error code for a failed system call. Look up the UNIX error. Contact the Cisco Technical Assistance Center.

* POL323 Child process: system() call \"command name"\ failed, errno=<error number>

The <command name> argument represents the name of the command on which the system call failed, and the <error number> argument represents the UNIX error number. The child POL task displays an error code for a failed system call. Look up the UNIX error. Contact the Cisco Technical Assistance Center.

* POL325 fork() failed

The system reports a fork failure for the named site. This message indicates the task cannot create a child process. Restart the application. If the problem persists, contact the Cisco Technical Assistance Center.

* POL326 Login failed for VSC unit <MGC node name>, please check ID/passwd

The <MGC node name> argument represents the MGC node name. The login has failed for the Cisco MGC unit. Check the user ID and password. Contact the Cisco Technical Assistance Center.

* POL327 Login failed for VSC unit <MGC node name>, please check ID/passwd

The <MGC node name> argument represents the name of the Cisco MGC node. The system was unable to log in to the named Cisco MGC unit. Check the user ID and password. Check **setpollpwd**. Contact the Cisco Technical Assistance Center. To clear this alarm, you must provide a matching text message, because multiple POL327 alarms can be active (each with a different message).

Note   This alarm is cleared after the alarm condition is removed.

* POL328 File <filename> send to redundant poll unit failed

The <filename> argument represents the name of the file intended to be sent. The system was unable to send the named file to the redundant unit. Check that FTP is working on the BAMS unit. Create a test file on the Cisco MGC and manually FTP or SFTP it to the BAMS unit. If the attempt is unsuccessful, there might be a network problem. Contact the Cisco Technical Assistance Center.

POL329 File <filename> sent to redundant poll unit successfully

The <filename> argument represents the name of the file sent. In a redundant system, after a Cisco MGC file is polled by one BAMS unit, it is processed by that unit and is also sent to the redundant BAMS unit for processing. This is an informational message. No action is necessary.

POL330 Cannot open XFR directory, errno <error number>

The <error number> argument represents the error number. The polling task cannot open the XFR directory to transfer files to the redundant BAMS unit. Check the error. If the problem persists, contact the Cisco Technical Assistance Center.

* POL331 Polling unit rotates from unit <from unit number> to <to unit number> failed
The <from unit number> argument represents the polling unit that was actively polling just before the unit rotation and the <to unit number> argument represents the unit that was on standby. The system was unable to perform the requested polling unit rotation. Check the network connection between the BAMS units. Perform an FTP or SFTP test. Contact the Cisco Technical Assistance Center.

* POL332 Polling unit rotates from unit <from unit number> to <to unit number>, but CTLFILE transfer failed
The <from unit number> argument represents the polling unit that was actively polling just before the unit rotation and <to unit number> represents the unit that was on standby. The POL task was able to perform the requested polling unit rotation, but it was unable to transfer the necessary control file information. Check the network and perform an FTP or SFTP test. Contact the Cisco Technical Assistance Center.

* POL333 Error in linking to <directory name> directory, errno <error number>
The <directory name> argument represents the directory name. The <error number> argument represents the UNIX error code. The POL task was unable to perform the requested file linking operation. Check the permissions for the specified directory. Contact the Cisco Technical Assistance Center.

* POL334 Cannot attach to shared memory, errno <error number>
The <error number> argument represents the UNIX error code. The POL task was unable to access the shared device memory. Use the ipcs utility to check the shared memory status. Contact the Cisco Technical Assistance Center.

* POL335 Rotate cannot be executed in non-polling unit
The POL task was unable to rotate the requested unit that was on standby at the time. The system only allows polling unit rotation from the current polling unit to the standby unit.

* POL336 File <filename> received from unit <from unit number> failed
The <filename> argument represents the name of the file that the system was unable to receive and the <from unit number> argument represents the unit number of the sending device. The POL task was unable to receive a file from the named unit and site. Perform an FTP or SFTP test. Contact the Cisco Technical Assistance Center.

* POL338 Site number invalid or not specified
The POL task is unable to validate the site number; it might be undefined. Check polling parameters, and ensure that the site exists. Contact the Cisco Technical Assistance Center.

* POL340 No data in 3 consecutive polling sessions
The POL task was unable to execute the requested polling operation in three consecutive attempts. Check the Cisco MGC and verify the existence of data. Use the **chgno** command to check the sequence numbers on the BAMS units. Contact the Cisco Technical Assistance Center.

* POL341 Cannot access shared memory, MGR may not have started
The polling tasks cannot get the contents of the shared memory. If the problem persists, contact the Cisco Technical Assistance Center.

* POL342 Cannot <access> xfr directory <directory name>, errno <error number>
The <access> argument represents the operation, <directory name> represents the name of the directory, and <error number> represents the error number. The polling task is unable to perform the named operation on the xfr directory. Check the error message. If the problem persists, contact the Cisco Technical Assistance Center.

* POL343 Cannot <operation> site data directory <directory>, errno <error number>
The <operation> argument represents the attempted system operation, <directory> represents the site data directory, and <error number> represents the UNIX error number. The POL task is unable to perform the specified system operation on the site data directory and displays a UNIX error code. Look up the error. Check the permissions of the BAMS user to access or create this directory. Contact the Cisco Technical Assistance Center.

* POL353 Cannot read record in file <control filename>.CTL
The <control filename> argument represents the control filename. The POL task was unable to access the requested record. The record might be locked by another process. Determine what other process is using and locking the record. Contact the Cisco Technical Assistance Center.

* POL356 Record already in use in file <control filename>.CTL
The <control filename> argument represents the control filename. This record is already locked by another process. Try again later. If the problem persists, contact the Cisco Technical Assistance Center.

* POL358 Invalid REDUNDANCY_MODE
The symbol definitions contain an invalid redundancy mode specification. Check the sym_defs configuration file in the opt/CiscoBAMS/bin directory. The Redundancy mode value should be 2 for redundant and 1 for simplex configuration. Run the **change_mode** command in the opt/CiscoBAMS/bin directory if the setting is wrong. Contact the Cisco Technical Assistance Center.

* POL359 Cannot rotate unit in simplex mode
The POL task is unable to rotate polling units, because the system in operating in simplex (single-unit) mode, which does not permit a polling unit rotation. Contact the Cisco Technical Assistance Center.

* POL360 Open control file <filename> failed
The <filename> argument represents the name of the control file. The POL task is unable to open the control file. Check permissions of the control file and ownership. Contact the Cisco Technical Assistance Center.

* POL361 Polling config record is blank in <filename>
The <file name> argument represents the name of the control file. The polling configuration record is blank. Check the polling configuration file and verify that polling configuration file is populated for this site. Contact the Cisco Technical Assistance Center.

> ✎
> **Note**   This alarm is cleared after the alarm condition is removed.

* POL362 Sequence number is blank for config record in <filename>.CTL
The <filename> argument represents the name of the control file. The sequence number is blank. Use the **chgno** command to specify the sequence number to poll. Contact the Cisco Technical Assistance Center.

* POL363 Cannot open <directory name> directory, errno <error number>
The <directory name> argument represents the name of the polled directory, and <error number> represents the UNIX error. The POL task cannot open the named directory and returns a UNIX error number. Check the UNIX error number and permissions on the directory.

* POL364 <function name> error for CTL file <filename>: <error message>
The <function name> argument represents the name of the webapi function, <filename> represents the name of the CTL file, and <error message> represents the error message. The webapi function returns an error for the displayed CTL file. Contact the Cisco Technical Assistance Center.

*C POL401 Max FTP failures for one file reached
The POL task has exceeded the maximum FTP failure threshold specified for a single file polling operation. Contact the Cisco Technical Assistance Center *immediately* and request that personnel there check the network and perform an FTP or SFTP test.

> ✎
> **Note**   This alarm is cleared after the alarm condition is removed.

```
*C POL402 Cannot connect to unit <unit name>
```
The <unit name> argument represents the name of the unit the system is attempting to poll. The system is unable to connect to the specified unit. Contact the Cisco Technical Assistance Center *immediately* and request that personnel there check the network and perform an FTP or SFTP test. To clear this alarm, you must provide a matching text message, because multiple POL402 alarms can be active (each with a different message).

Note   This alarm clears after the alarm condition is removed, or the system connects to either the principal (hostname) or the backup (hostname_b) site.

```
*C POL403 Standby/Standby state detected. BAMS unit <unit number> will go active, seqno1:
<sequence number 1> seqno2: <sequence number 2>
```
The <unit number> represents the unit number of the BAMS unit, <sequence number 1> represents the first sequence number, and <sequence number 2> represents the second sequence number. If a standby/standby condition is detected where both BAMS units are in the standby state for three consecutive intervals, the POL task of BAMS unit 0 goes active. This message is generated to indicate this condition. No action is required. BAMS has automatically recovered from a problem condition (standby/standby state).

Control File Receive Task Messages

The Control File Receive (RCV) task receives control files. During processing, this task writes messages to the syslog with an RCV task identifier (for example, RCV001).

```
*RCV501 Error loading record <error number>
```
The <error number> argument represents the UNIX error code. The RCV task is unable to load a control record. Look up the error number. Contact the Cisco Technical Assistance Center.

```
* RCV502 Invalid operation <operation number>
```
The <operation number> argument represents the operation number. The RCV task has performed an invalid operation. Contact the Cisco Technical Assistance Center.

```
* RCV503 Cannot open <filename>
```
The <filename> argument represents the name of the file that could not be opened. The RCV task is unable to open the named file. Verify that the file exists and that the file directory can be written to by user bams. Contact the Cisco Technical Assistance Center.

```
* RCV507 FTP failed for <filename>
```
The <filename> argument represents the name of the file that the RCV task could not operate on. The RCV task is unable to perform an FTP or SFTP operation on the named file. Check if the control file name exists on the remote BAMS unit. Check the network connection to the remote BAMS unit and perform an FTP or SFTP test. Contact the Cisco Technical Assistance Center.

```
* RCV508 Cannot get file status of <filename>
```
The <filename> argument represents the name of the file. The RCV task is unable to get a file status for the named file. Check if the control file exists on the local BAMS unit and if the user has read permissions on the file. Contact the Cisco Technical Assistance Center.

```
* RCV509 File size incorrect for FTP <filename>
```
The <filename> argument represents the name of the file. The RCV task is unable to match the file size for the named file. Perform an FTP or SFTP test. If the file sizes match, contact the Cisco Technical Assistance Center.

* RCV520 Cannot connect to unit <unit host name>
The <unit host name> argument represents the name of the unit host. The RCV task is unable to connect to the other unit. Check the network connection to <unit host name>. Contact the Cisco Technical Assistance Center.

* RCV521 Cannot open directory file <directory name>
The <directory name> argument represents the name of the directory. The RCV task is unable to open the named directory file. Check if the user has permission to read directory <directory name>. Contact the Cisco Technical Assistance Center.

* RCV522 system () call failed, errno = <error number>
The <error number> argument represents the UNIX error number. A system call has failed and the system displays a UNIX error. Look up the error number. Contact the Cisco Technical Assistance Center.

* RCV524 Max FTP failures for one file reached
The RCV task has failed to FTP a file after the maximum number of tries. Check the network connection to the remote BAMS unit and perform an FTP or SFTP test. Contact the Cisco Technical Assistance Center.

* RCV525 fork() failed
The RCV task experienced a fork failure at the named site. Contact the Cisco Technical Assistance Center.

* RCV527 Login failed for unit <unit host name>, please check ID/passwd
The <unit host name> argument represents the name of the unit host. The login has failed for the other BAMS unit. Check the user ID and password using **setbamunit**. Contact the Cisco Technical Assistance Center.

RCV528 Receive control information from redundant poll unit host <unit host 1> failed, trying host <unit host2>
The <unit host 1> argument represents the name of the first unit host, and <unit host 2> represents the name of the unit host for the second Ethernet card. The RCV task was unable to receive the control file from the displayed host. Check the network connection to the remote BAMS <unit host 1> and perform an FTP or SFTP test.

Send File Transfer Task Messages

The Send File Transfer (SXF) task sends control files to a redundant BAMS unit. During processing, this task writes messages to the syslog with an SXF task identifier (for example, SXF001).

* SXF037 Error loading control record, returned <return value>
The <return value> argument represents the system-returned error value. The SXF task experienced an error loading a control record and returned an error value. Contact the Cisco Technical Assistance Center.

SXF041 Send control information to redundant poll unit host <unit host name> failed, trying host <unit host name of second ethernet card>
The <unit host name> argument represents the first unit host name polled, and <unit host name of second ethernet card> represents the unit host name of the second Ethernet card. The SXF task failed in its attempt to send the control record and is trying another line. Check the network connection to the remote BAMS <unit host name> and perform an FTP or SFTP test.

* SXF502 stat () on <filename> failed, errno <error number>
The <filename> argument represents the name of the file, and <error number> represents the UNIX error number. Check the UNIX error.

```
* SXF503 <filename> size incorrect: remote=<remote size>, local=<standard buffer size>
```
The <filename> argument represents the name of the file, <remote size> represents the size on the remote unit, and <standard buffer size> represents the standard buffer size on the local unit. Check the network status. A communication problem exists between the two BAMS units. Check whether the Ethernet connection is good. Verify that the setbamunit utility has the correct parameters.

```
* SXF510 Attempt #<number> to rename or transfer <filename> failed
```
The <number> argument represents the number of communication attempts, and <filename> represents the name of file to be transferred or renamed. Check the network status. A communication problem exists between the two BAMS units. Check whether the Ethernet connection is good. Verify that the setbamunit utility has the correct parameters.

```
* SXF640 <directory filename> No such file or directory.
```
The <directory filename> argument represents the directory file name. Verify the path exists on the remote unit.

UNIX Error Numbers

Table A-16 lists the UNIX error numbers that might appear within system messages that BAMS tasks write to the syslog.

Table A-16   UNIX Error Numbers

Number	Reason
1	Not super-user
2	No such file or directory
3	No such process
4	Interrupted system call
5	I/O error
6	No such device or address
7	Arg list too long
8	Exec format error
9	Bad file number
18	Cross-device link
19	No such device
20	Not a directory
21	Is a directory
22	Invalid argument
23	File table overflow
24	Too many open files
25	Inappropriate ioctl for device
26	Text file busy
27	File too large
28	No space left on device
29	Illegal seek

Table A-16   UNIX Error Numbers (continued)

Number	Reason
30	Read only file system
31	Too many links
32	Broken pipe
33	Math argument out of domain of function
34	Math result not representable
35	No message of desired type
36	Identifier removed
37	Channel number out of range
38	Level 2 not synchronized
39	Level 3 halted
40	Level 3 reset
41	Link number out of range
42	Protocol driver not attached
43	No CSI structure available
44	Level 2 halted
45	Deadlock condition
46	No record locks available
47	Operation canceled
48	Operation not supported
49	Disk quota exceeded
50	Invalid exchange
51	Invalid request descriptor
52	Exchange full
53	No anode
54	Invalid request code
55	Invalid slot
56	File locking deadlock error
57	Bad font file format
60	Device not a stream
61	No data (for no delay I/O)
62	Timer expired
63	Out of streams resources
64	Machine is not on the network
65	Package not installed
66	The object is remote
67	The link has been severed

Table A-16   UNIX Error Numbers (continued)

Number	Reason
68	Advertise error
69	Srmount error
70	Communication error on send
71	Protocol error
74	Multihop attempted
77	Trying to read unreadable message
78	Path name is too long
79	Value too large to be stored in data type
80	Given log. name not unique
81	F.D. invalid for this operation
82	Remote address changed
83	Cannot access a needed shared library
84	Accessing a corrupted shared library
85	.lib section in a.out corrupted
86	Attempting to link in too many libraries
87	Attempting to execute a shared library
88	Illegal byte sequence
89	Unsupported file system operation
90	Symbolic link loop
91	Restartable system call
92	If pipe/FIFO, do not sleep in stream head
93	Directory not empty
94	Too many users (for UFS)
95	Socket operation on non-socket
96	Destination address required
97	Message too long
98	Protocol wrong type of socket
99	Protocol not available
120	Protocol not supported
121	Socket type not supported
122	Operation not supported on socket
123	Protocol family not supported
124	Address family not supported by protocol family
125	Address already in use
126	Cannot assign requested address
127	Network is down

Table A-16   UNIX Error Numbers (continued)

Number	Reason
128	Network is unreachable
129	Network dropped connection because of reset
130	Software caused connection abort
131	Connection reset by peer
132	No buffer space available
133	Socket is already connected
134	Socket is not connected
143	Cannot send after socket shutdown
144	Too many references: can't splice
145	Connection timed out
146	Connection refused
147	Host is down
148	No route to host
149	Operation already in progress
150	Operation now in progress
151	Stale NFS file handle

Signal Numbers

displays the signal numbers that might appear within system messages that BAMS tasks write to the syslog.

Table A-17   Signal Numbers

Number	Name	Reason
1	SIGHUP	Hangup
2	SIGINT	Interrupt (rubout)
3	SIGQUIT	Quit (ASCII FS)
4	SIGILL	Illegal instruction (not reset when caught)
5	SIGTRAP	Trace trap (not reset when caught)
6	SIGIOT	IOT instruction
7	SIGEMT	EMT instruction
8	SIGFPE	Floating-point exception
9	SIGKILL	Kill (cannot be caught or ignored)
10	SIGBUS	Bus error
11	SIGSEGV	Segmentation violation
12	SIGSYS	Bad argument to system call

Table A-17   Signal Numbers (continued)

Number	Name	Reason
13	SIGPIPE	Write on a pipe with no one to read it
14	SIGALRM	Alarm clock
15	SIGTERM	Software termination signal from kill
16	SIGUSR1	User-defined signal 1
17	SIGUSR2	User-defined signal 2
18	SIGCLD	Child status change
18	SIGCHLD	Child status change alias (POSIX)
19	SIGPWR	Power-fail restart
20	SIGWINCH	Window size change
21	SIGURG	Urgent socket condition
22	SIGPOLL	Pollable event condition
22	SIGIO	Socket I/O possible (SIGPOLL alias)
23	SIGSTOP	Stop (cannot be caught or ignored)
24	SIGSTP	User stop requested from tty
25	SIGCONT	Stop process has been continued
26	SIGTTIN	Background tty read attempted
27	SIGTTOU	Background tty write attempted
28	SIGVTALRM	Virtual time expired
29	SIGPROF	Profiling timer expired
30	SIGXCPU	Exceeded CPU limit
31	SIGXFSZ	Exceeded file size limit
32	SIGWAITING	Process's lwps are blocked
33	SIGLWP	Special signal used by thread library
34	SIGFREEZE	Special signal used by CPR
35	SIGTHAW	Special signal used by CPR
36	SIGCANCEL	Thread cancellation signal used by libthread
37	SIGTMIN	First (highest-priority) real-time signal
44	SIGRTMAX	Last (lowest-priority) real-time signal

Signal Numbers