Log Central

Configuration Guide

BEA Log Central 5.0
Document Edition 5.0
October 2000

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Log Central Configuration Guide

Document Edition	Date	Software Version
5.0	October 2000	Log Central 5.0

Contents

About This Document

What You Need to Know viii
e-docs Web Site viii
How to Print the Document ix
Related Information ix
Contact Us! ix
Documentation Conventions x

1. Overview

Log Messages in System Management 1-2
Log Messages in Log Central 1-2
Log Central Architecture 1-3
    Overview of Log Central Architecture 1-3
    Data Collection Agent 1-5
    Log Monitor 1-6
    Message Sender 1-6
    Central Collector 1-7
    Message Receiver 1-8
    Message Processor 1-8
Construction of Log Messages 1-8
Log Central Processes 1-9
    Overview of the Log Central Processes 1-10
    Process Monitor 1-11
SNMP Integration 1-12
Fault Tolerance 1-12
Log Central Console 1-13

2. Setting Up Log Central

Installing Log Central 2-1
Configuring Log Central 2-2
Starting Log Central 2-3

3. Configuring the Central Host

4. Configuring Multiple Instances of Log Central

When To Configure Multiple Instances 4-1
Configuring Multiple Instances 4-2

5. Creating Log Mappings

What Is a Log Mapping? 5-2
Where to Define Log Mappings 5-3
Using Multiple Log Mappings 5-3
Mapping Options 5-3
Date Formats for the -D Option 5-5
Specifying Option Values 5-6
Mapping Dates 5-8
Working with Metacharacters 5-9
Specifying Multiple Separators 5-10
Working with Field Lengths 5-11
Example: Mapping a System Log 5-12

6. Creating Message Definitions

Overview of Message Definitions 6-1
Creating a Message Definition File 6-3

7. Creating Filters

Overview of Filters 7-1
Defining Filters 7-2
    Creating a DEFINE_FILTER Entry 7-2
    Defining a Condition 7-3
    Filtering Based on a Domain 7-5
    Defining an Action 7-6
    Suppressing an Action 7-7

Assigning Filters 7-9
Overview of Assigning Filters 7-9
Assigning a Global Filter 7-9
Turning Off a Global Filter 7-10
Assigning a Local Filter 7-10

8. Integrating SNMP

Overview of SNMP and MIBs in Log Central 8-1
Setting Up SNMP Management 8-2
What Traps Are Generated? 8-3

A. Log Central Message and Message Definition Formats

Example of a Log Central Message and Message Definition A-2
Message Header A-3
Log ID A-4
Logging Level A-4
Date and Time A-5
Subsystem A-5
Message ID A-6
Host A-6
Process ID A-7
User ID A-7
Function A-8
Entity A-9
Message Body A-9
Message Definition A-10
Summary A-11
Severity A-11
Description A-12
Recommendation A-12
Execute on Upload A-13
Trap ID A-13
Trap Enabled A-14
Auto Acknowledge A-14

B. Commands

lc_config B-2
lc_create_schema B-4
lc_drop_schema B-4
lc_launch B-5
lc_user_create B-6
lc_user_delete B-7
lc_user_modify B-8
lc_user_list B-9
log_monitor B-10
msgdef_delete B-13
msgdef_export B-14
msgdef_import B-15
msg_reader B-15
msg_test B-16
show_config B-17
start_messaging B-19
stop_messaging B-21
subsystem_create B-21
subsystem_delete B-22

C. Environment Variables

Setting an Environment Variable C-2
BEA_LC_CONF_SERVICE C-2
BEA_LC_HOST_CONF C-2
BEA_LC_IPCKEY C-3
BEA_LC_MONITOR_WAKEUP_INTVL C-3
BEA_LC_PROC_MAX_RESTARTS C-3
BEA_LC_PROC_RESTART_INTVL C-4
BEA_LC_TALK_SERVICE C-4
BEA_LC_TRAP_CONF C-4

D. MIB Reference

Overview of MIBs D-1
beaTrap: Log Central Traps MIB D-1

beaTrapLcLogLevel D-2
beaTrapLcTimestamp D-3
beaTrapLcSubsys D-3
beaTrapLcMid D-4
beaTrapLcHost D-4
beaTrapLcPid D-4
beaTrapLcUid D-5
beaTrapLcFunction D-5
beaTrapLcTxKey D-5
beaTrapLcVersion D-6
beaTrapLcSeverity D-6
beaTrapLcMessageBody D-6

E. Database Schema

IL_MSG: Message Definition E-1
IL_REP_MODE: Logging Level Definitions E-3
IL_SEV: Severity Level Definitions E-3
IL_SUBS: Subsystem Definitions E-4
IL_SM_LOG_TABLE: Message Header Definition E-4
IL_TRAP_CLASS: Trap Classes E-6
IL_USER: User Data E-6

F. Initialization File

G. Configuration Files

Messaging Configuration File G-1
Overview of the Messaging Configuration File G-2
Example Messaging Configuration File G-2
LC_GLOBAL G-4
MANAGED_NODE G-6
DEFINE_FILTER G-7
DEFINE_MSG_MAPPINGS G-7
DEFINE_LOG_MONITOR G-8
Trap Configuration File G-9
Overview of the Trap Configuration File G-9
Example Trap Configuration File G-10

TRAP_HOST G-10

H. Predefined Log Mappings

Names and Commands for Predefined Log Mappings.................................... H-2
BEA Tuxedo Log and BEA WebLogic Enterprise Log H-2
LM Predefined Mapping H-3
Oracle Alert Log... H-4
Windows NT Event Log... H-5

Index

About This Document

This document describes the BEA® Log Central application and explains how to configure Log Central.

This document is organized as follows:

- Chapter 1, “Overview,” describes the Log Central architecture.
- Chapter 2, “Setting Up Log Central,” explains how to install, configure, and start Log Central.
- Chapter 3, “Configuring the Central Host,” explains how to configure the central host.
- Chapter 4, “Configuring Multiple Instances of Log Central,” explains how to configure multiple instances of Log Central.
- Chapter 5, “Creating Log Mappings,” explains how to create log mappings.
- Chapter 6, “Creating Message Definitions,” explains how to create message definitions.
- Chapter 7, “Creating Filters,” explains how to create filters.
- Chapter 8, “Integrating SNMP,” explains how to integrate SNMP into Log Central.
- Appendix A, “Log Central Message and Message Definition Formats,” describes the Log Central message format.
- Appendix B, “Commands,” describes the Log Central commands.
- Appendix C, “Environment Variables,” describes the environment variables that Log Central uses.

- Appendix D, “MIB Reference,” describes the MIB that the Log Central SNMP feature uses.
- Appendix E, “Database Schema,” describes the Log Central database schema.
- Appendix F, “Initialization File,” describes the Log Central initialization file.
- Appendix G, “Configuration Files,” describes the Log Central messaging configuration file and trap configuration file.
- Appendix H, “Predefined Log Mappings,” describes the predefined log mapping that Log Central provides.

What You Need to Know

This document is written for system administrators and network administrators who set up and manage Log Central. It assumes a working knowledge of relational databases and JDBC. If you use the SNMP feature, then you also need to have a working knowledge of SNMP.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click Product Documentation or go directly to the e-docs Product Documentation page at http://e-docs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the Log Central documentation Home page on the e-docs Web site and also on the Log Central product CD. You can open the PDF in Adobe Acrobat Reader and print the entire document or a portion of it in book format. To access the PDFs, open the Log Central documentation Home page, click the PDF files button and select the document to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

For more information about Log Central, see the Log Central online documentation set which is available on the Log Central product CD and at the e-docs Product Documentation page at http://e-docs.bea.com/.

Contact Us!

Your feedback on the BEA Log Central documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the Log Central documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA Log Central 5.0 release.

If you have any questions about this version of BEA Log Central, or if you have problems installing and running BEA Log Central, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

Convention	Item

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item

`...`	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name] [-f `*`file-list`*`]...`
	`[-l `*`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Overview

The following sections provide an overview of Log Central:

- Log Messages in System Management
- Log Messages in Log Central
- Log Central Architecture
    - Overview of Log Central Architecture
    - Data Collection Agent
    - Log Monitor
    - Message Sender
    - Central Collector
    - Message Receiver
    - Message Processor
- Construction of Log Messages
- Log Central Processes
    - Overview of the Log Central Processes
    - Process Monitor
- SNMP Integration
- Fault Tolerance
- Log Central Console

Log Messages in System Management

Log messages are typically used as a system management tool: to detect problems, track down the source of a fault, or track system performance. Distributed systems include a variety of software components that generate message logs, such as operating systems and relational database management systems (RDBMS). In the absence of an industry-wide standard, software makers use different practices for message logging.

Log Messages in Log Central

Log Central extracts information from different kinds of logs and maps the information into a common format. Log Central stores the reformatted log messages in a relational database, which gives you a single point of access and a unified view of the log messages. Thus, Log Central makes it easier for you to manage your distributed systems.

A single failure, such as a file system filled to capacity, can generate multiple log messages as the problem ripples through the affected software. A unified view of the log messages lets you diagnose the source of a problem more rapidly.

Log Central helps you manage BEA products such as BEA Tuxedo®, BEA WebLogic Enterprise™, and BEA WebLogic Server™. You can also use Log Central to manage databases, operating systems, and any other software programs that generate log messages.

Log Central Architecture

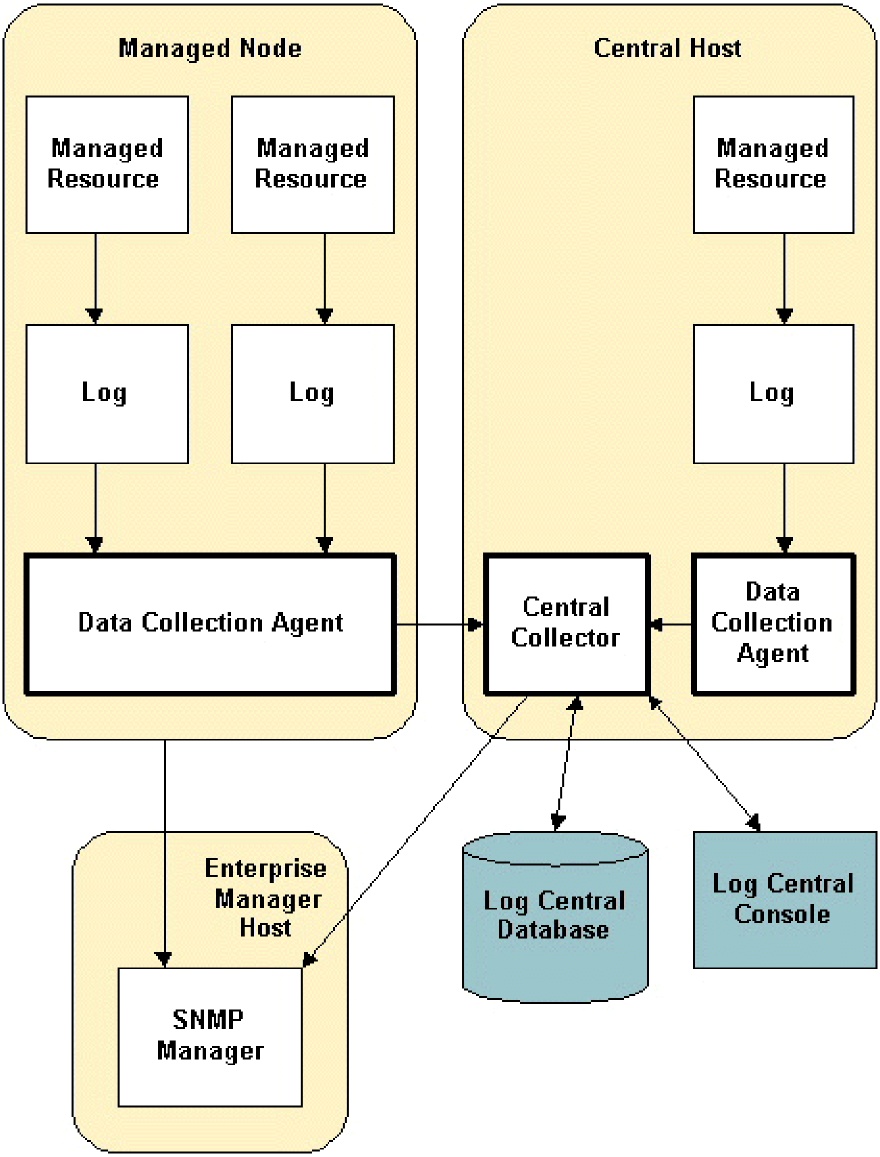
The following sections describe the Log Central architecture:

- Overview of Log Central Architecture
- Data Collection Agent
- Log Monitor
- Message Sender
- Central Collector
- Message Receiver
- Message Processor

Overview of Log Central Architecture

Log Central is based on an agent/manager architecture as shown in the following figure. A *managed node* is a machine that has resources that need to be managed. A *managed resource* is a software component such as a BEA Tuxedo application, an operating system, or an RDBMS. The central host and each managed node has a Log Central Data Collection Agent. The Data Collection Agents forward log messages to the Central Collector, which stores log messages in a database and provides information to the Log Central Console. Data Collection Agents and the Central Collector can generate SNMP traps. The Log Central Console, the Central Collector, and the Log Central database play the manager role in the Log Central architecture.

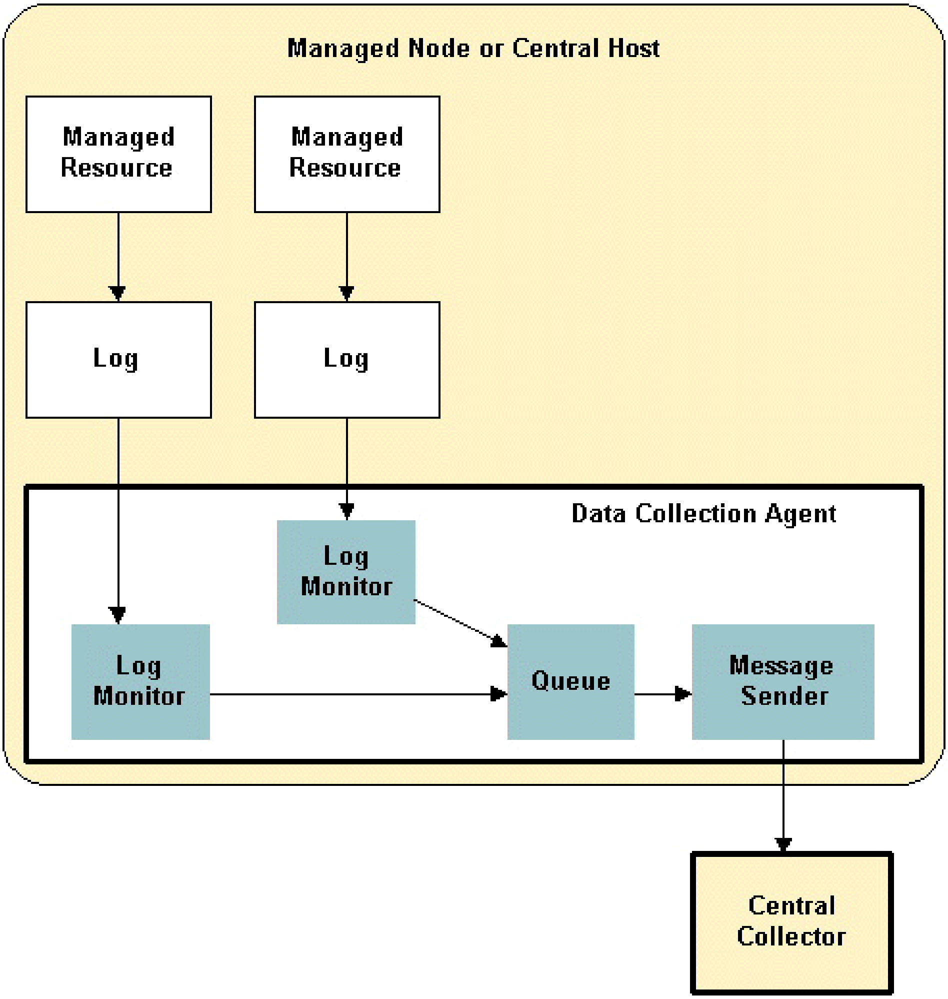
Figure 1-1 Log Central Architecture



Data Collection Agent

A Data Collection Agent resides on a managed node or the central host. A Data Collection Agent consists of a Message Sender, a queue, and one or more Log Monitors. The following figure shows the flow of log information from managed resources through a Data Collection Agent to the Central Collector.

Figure 1-2 Data Collection Agent



Log Monitor

A Data Collection Agent needs a Log Monitor for each managed resource. The Log Monitor receives log messages from the managed resource and maps these messages into the Log Central message format before putting them in the queue. For information about mapping, see Chapter 5, “Creating Log Mappings.”

Message Sender

The Message Sender reads messages from the queue and forwards them to the Central Collector. You can create filters that direct the Message Sender to do the following:

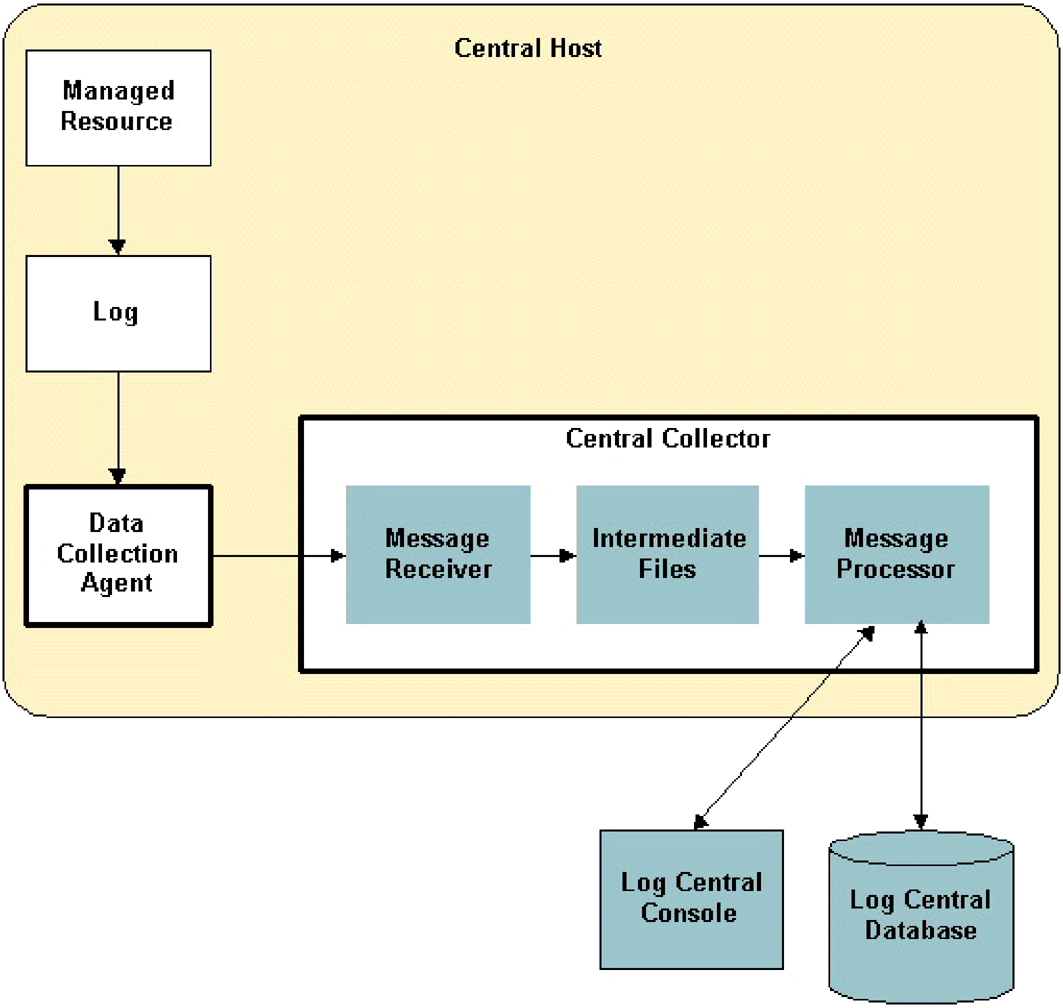
- Discard (not forward) specified messages.
- Save specified message to a file.
- Send an SNMP trap when a specified message is received.
- Run a script or program when a specified message is received.

For information about creating filters, see Chapter 7, “Creating Filters.”

Central Collector

The Central Collector resides on the central host and consists of a Message Receiver, intermediate files, and a Message Processor. The following figure shows the flow of log information in the Central Collector.

Figure 1-3 Central Collector



Message Receiver

After receiving a log message from a Data Collection Agent, the Message Receiver puts the message in an intermediate file. The Message Receiver creates a new intermediate file every hour. Use the Storage Maintenance tool, which is in the Log Central Console, to control how frequently the Message Receiver deletes intermediate files.

Message Processor

The Message Processor performs the following functions:

- Retrieves log messages from the intermediate files and puts them in the Log Central database
- Handles requests from the Log Central Console, such as database queries or forwarding of incoming messages for online monitoring
- Generates SNMP trap notifications for log messages that are mapped to SNMP traps

Construction of Log Messages

Log Central constructs log messages as described in the following steps and diagram:

1. When a Log Monitor receives a log message, it maps the message header and message body into the Log Central message format. Then the Log Monitor forwards the message to the Message Sender, which forwards the message to the Central Collector.

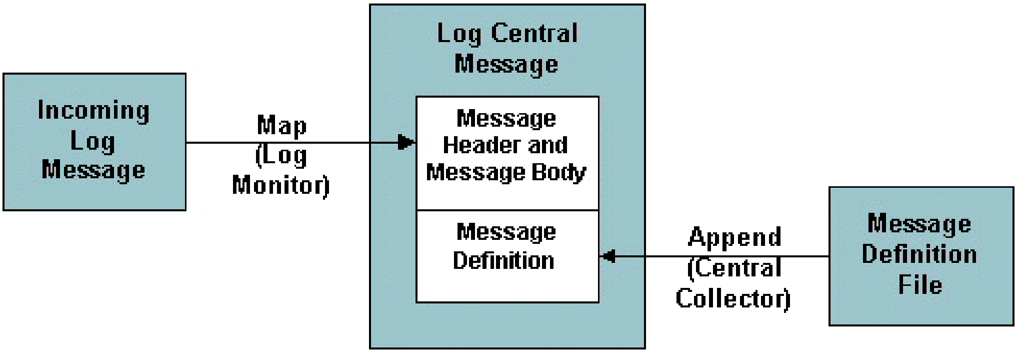
   For information about how the Log Monitor maps messages, see Chapter 5, “Creating Log Mappings.” For information about the Log Central message format, see Appendix A, “Log Central Message and Message Definition Formats.”

2. When the Central Collector receives the reformatted log message from the Message Sender, it uses the message's subsystem and message ID fields to find a corresponding message definition in a message definition file. Message definition files are in the Log Central database.

   For information about message definitions, see Chapter 6, "Creating Message Definitions," and Appendix A, "Log Central Message and Message Definition Formats."

3. The Central Collector appends the message definition to the message.

Figure 1-4 Construction of Log Messages



Log Central Processes

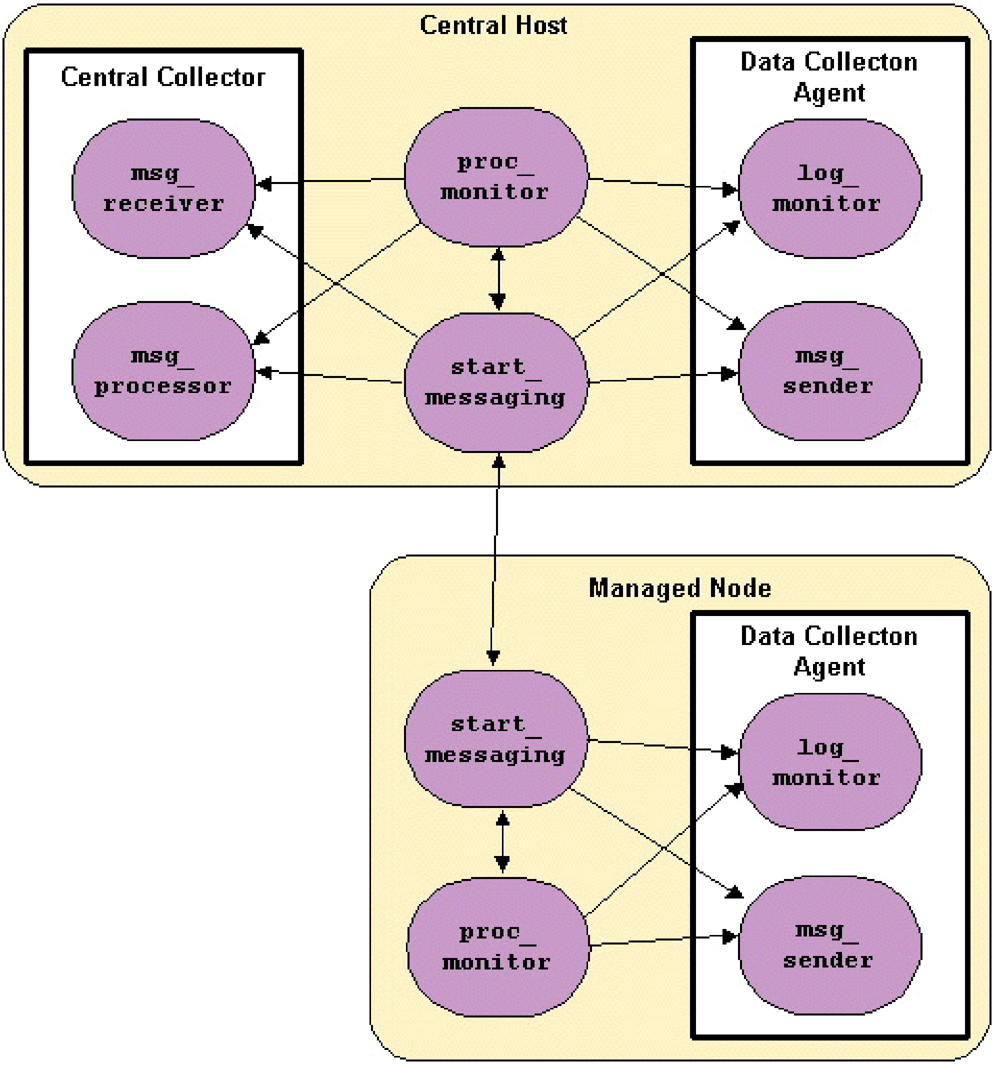
The following sections describe the Log Central processes:

- Overview of the Log Central Processes
- Process Monitor

Overview of the Log Central Processes

The following figure shows the Log Central processes.

Figure 1-5 Log Central Processes



When you run the `start_messaging` process on the central host, it starts the following processes:

- Message Receiver (`msg_receiver`)
- Message Processor (`msg_processor`)
- Process Monitor (`proc_monitor`)
- Message Sender (`msg_sender`)
- Log Monitors (`log_monitor`) if you included `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file.

When you call the `start_messaging` process on a managed node, it starts the following processes:

- Message Sender (`msg_sender`)
- Process Monitor (`proc_monitor`)
- Log Monitors (`log_monitor`) if you included `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file.

The `start_messaging` process on a managed node connects to the `start_messaging` process on the central host.

Process Monitor

The Process Monitor is a daemon that runs on the central host and each managed node. The following processes register with the Process Monitor at startup:

- `start_messaging`
- Log Monitor (`log_monitor`)
- Message Sender (`msg_sender`)
- Message Receiver (`msg_receiver`)
- Message Processor (`msg_processor`)

At fixed intervals, the Process Monitor checks all registered processes. If configured to do so, it restarts dead processes with the user and group IDs that were passed to it at startup.

SNMP Integration

Log Central enables you to use Simple Network Management Protocol (SNMP) to integrate information from logs into an enterprise management system. You can specify the following kinds of traps:

- Basic traps

  You use the Basic Trap Configuration window, which is part of the Log Central Console, to map log messages to SNMP traps. The Central Collector generates an SNMP trap when it receives a log message that is mapped to an SNMP trap.

- Advanced traps

  You can create filters in the Log Central messaging configuration file to specify advanced criteria for triggering SNMP traps. A Data Collection Agent generates an SNMP trap when it receives a message that matches the criteria. For information about creating filters, see Chapter 7, “Creating Filters.”

Fault Tolerance

You can configure a secondary Central Collector as a backup for the primary Central Collector. If the primary Central Collector becomes unavailable, the Data Collection Agents automatically send the log messages to the secondary Central Collector. Control automatically switches back to the primary Central Collector when it becomes available. The primary Central Collector can access the messages that were sent to the secondary Central Collector if both collectors use the same database. For information about configuring a secondary Central Collector, see Chapter 3, “Configuring the Central Host.”

If none of the Central Collectors configured for a Data Collection Agent are accessible, the Data Collection Agents save the log messages to a temporary local file. When the Central Collector becomes available, the Data Collection Agents recover the messages from the temporary files and forward them to the Central Collector. During recovery, new incoming messages have the highest priority and recovered messages are forwarded when the Data Collection Agents are not handling new messages. The Data Collection Agents delete the temporary files after the messages have been forwarded.

Log Central Console

The Log Central Console is a graphical user interface that lets you access the log messages that the Central Collector receives. You can use the Log Central Console to analyze problems and track trends. For more information about the console, see the Log Central Online Help.

CHAPTER

2 Setting Up Log Central

The following sections outline the basic steps for installing, configuring, and starting Log Central:

- Installing Log Central
- Configuring Log Central
- Starting Log Central

Installing Log Central

To install Log Central:

1. Install Log Central on the central host and managed nodes as described in the *BEA Log Central Installation Guide*.
2. Install the Log Central relational database.
   - For a list of supported databases, see the *BEA Log Central Release Notes.* You can install the database on the central host or on another machine in your network. To install the database, see the database vendor's documentation.
   - The database must have a Java Database Connectivity (JDBC) driver. To install the JDBC driver, see the database vendor's JDBC documentation. When you run `lc_config` to configure the central host, Log Central copies the JDBC driver to the *`install_dir`*`/bin` directory, where *`install_dir`* is the directory where you installed Log Central.

Configuring Log Central

To configure Log Central:

1. Create the Log Central database user.
   - Log Central uses this database user to access the Log Central database. This user needs to have privileges to create and delete tables and modify their contents.
   - Allocate a minimum of 10 MB of disk space for the database user.
   - To create the user, see the database vendor's documentation. You can also use the `lc_user_create` command, which is described in Appendix B, "Commands."
2. Configure the central host. See Chapter 3, "Configuring the Central Host."
3. If necessary, configure multiple instances of Log Central. See Chapter 4, "Configuring Multiple Instances of Log Central."
4. Configure the Log Central database.

   a. Run `lc_create_schema` on the central host to create a database schema.

      For the `lc_create_schema` syntax, see Appendix B, "Commands." If an error occurs or if `lc_create_schema` aborts, run `lc_drop_schema` to clean up files that may have been created, try to correct the problem, and re-create the schema.

   b. If desired, run `subsystem_create` to create additional subsystem entries in the database.

      The Log Central database is partitioned into different categories for the different types of applications and software components that Log Central monitors. Each category represents a resource that generates messages. These resources are called *subsystems*. The subsystem is one of the unique attributes in a log message.

Log Central automatically creates subsystems for the following types of messages:

- BEA Tuxedo logs
- BEA WebLogic Enterprise logs
- Oracle alert logs
- Windows NT event logs

If you are using Log Central to monitor additional types of applications or software components, call `subsystem_create` to create a subsystem for each additional type of log message. For the `subsystem_create` syntax, see Appendix B, "Commands."

5. Create log mappings. See Chapter 5, "Creating Log Mappings."
6. Create message definitions. See Chapter 6, "Creating Message Definitions."
7. If desired, integrate Log Central with SNMP. See Chapter 8, "Integrating SNMP."
8. If desired, set environment variables on the central host and managed nodes.

   You need to set environment variables if you want to run Log Central without accepting all its defaults, such as with multiple instances of Log Central or for debugging purposes. Environment variables are described in Appendix C, "Environment Variables."
9. If desired, create filters. See Chapter 7, "Creating Filters."

Starting Log Central

To start Log Central:

1. Start Log Central:

   a. Start the database server.

   b. Run `start_messaging` for each Central Collector.

   The Central Collectors provide the Data Collection Agents with configuration information. Therefore, you must start the Central Collectors before you start the Data Collection Agents. For information about the `start_messaging` command, see Appendix B, “Commands.”

c. Run `start_messaging` for each Data Collection Agent.

   If you included `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file, then `start_messaging` automatically starts the Log Monitors. For information about the messaging configuration file, see Appendix G, “Configuration Files.” For information about the `start_messaging` command, see Appendix B, “Commands.”

d. If necessary, run `log_monitor` for each managed resource.

   If you did not set the messaging configuration file to automatically start the Log Monitors in the previous step, you must start each Log Monitor manually. For information about the `log_monitor` command, see Appendix B, “Commands.”

2. Configure and start the Log Central Console.

   a. Modify `PATH` to include the directory that contains a Web browser. By default, the Log Central Console uses Netscape Navigator to display the online help. To use a non-default browser, set the `-b` option when you start the Log Central Console.

   b. Run `start_messaging` on the central host.

   c. Wait approximately one minute. This wait time allows the Log Central processes to start.

   d. Run `lc_launch` on the central host.

   For information about the `start_messaging` and `lc_launch` commands, see Appendix B, “Commands.” For information about the Log Central Console, see the Log Central Online Help.

3. To stop Log Central, run `stop_messaging` for each Central Collector and each Data Collection Agent.

   For information about the `stop_messaging` command, see Appendix B, “Commands.”

CHAPTER

3 Configuring the Central Host

The Log Central Host Configuration Utility helps you configure the central host. The utility creates the initialization file and the messaging configuration file. For information about these files, see Appendix F, “Initialization File,” and Appendix G, “Configuration Files.”

To configure the central host:

1. Determine the value for each field described in the following table. The values in the “Field #” column correspond to the callout numbers in Figure 3-1.

Table 3-1 Fields in the Log Central Host Configuration Utility

Field #	Field	Description
1	Central Host Name	Name of the central host. This is the value returned when you run `hostname` on the central host. (`hostname` is a standard DOS command and a standard UNIX command.) Equivalent value in the message configuration file: `CENTRAL_HOST`
2	Intermediate File Prefix	Directory and file prefix for the intermediate files that the Message Receiver creates. If the directory is on a remote file system, Log Central performance can be adversely affected. Equivalent value in the message configuration file: `LOGPREFIX`
3	Port Number	Port number on which the Message Processor listens for requests from the Log Central Console Equivalent value in the initialization file: LC.Server.dbPort

Table 3-1 Fields in the Log Central Host Configuration Utility

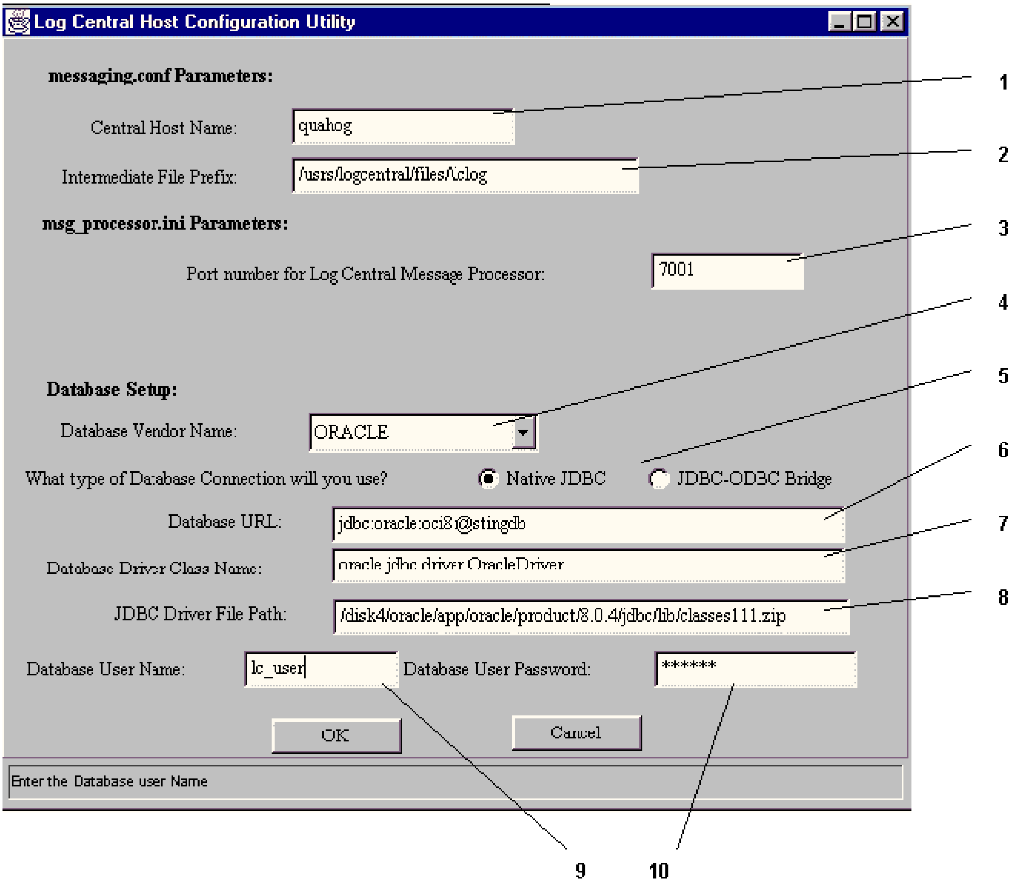
Field #	Field	Description

4	Database Vendor Name	Name of the database vendor. Possible values:
		`MSSQL`
		`ORACLE`
		Equivalent value in the initialization file: LC.DBVendor

5	Type of Database Connection	Native JDBC or JDBC-ODBC bridge. This selection specifies the default values that the Log Central Host Configuration Utility displays in the Database URL and Database Driver Class Name fields.
6	Database URL	URL for accessing the database. For more information, see your database vendor's documentation. Equivalent value in the initialization file: LC.URL
7	Database Driver Class Name	Name of the database driver class. For more information, see your database vendor's documentation. Equivalent value in the initialization file: LC.driver
8	JDBC Driver File Path	Path for the location of the JDBC driver. The Log Central Host Configuration utility copies the JDBC driver file to *`install_dir`*`/bin/JDBCDrvrForLC.zip`, where *`install_dir`* is the directory where you installed Log Central.
9	Database User Name	User that you created in step 2 of the Log Central setup procedure. The Log Central Host Configuration utility uses this user name to create and populate the tables in the Log Central database. Equivalent value in the initialization file: LC.userName
10	Database User Password	Password for the database user. Equivalent value in the initialization file: LC.password

2. Run the Log Central Host Configuration utility, `lc_config`, on the central host. For the command syntax, see Appendix B, “Commands.”

Figure 3-1 Log Central Host Configuration Utility



Log Central Host Configuration Utility

messaging.conf Parameters:

Central Host Name: quahog

Intermediate File Prefix: /usrs/logcentral/files/lclog

msg_processor.ini Parameters:

Port number for Log Central Message Processor: 7001

Database Setup:

Database Vendor Name: ORACLE

What type of Database Connection will you use? ● Native JDBC ○ JDBC-ODBC Bridge

Database URL: jdbc:oracle:oci8:@stingdb

Database Driver Class Name: oracle.jdbc.driver.OracleDriver

JDBC Driver File Path: /disk4/oracle/app/oracle/product/8.0.4/jdbc/lib/classes111.zip

Database User Name: lc_user Database User Password: *******

Enter the Database user Name

3. Enter the field values and click OK.

4. If desired, configure a secondary Central Collector.

   You can modify the messaging configuration file to create a secondary Central Collector. For information about the messaging configuration file, see

Appendix G, “Configuration Files.” In the messaging configuration file, the `LC_GLOBAL` entry contains values for the central host and all managed nodes. To specify a secondary Central Collector, add the following lines to the `LC_GLOBAL` entry:

```
BACKUP_HOST = hostname
BACKUP_LOGPREFIX = log_file_dir/prefix
```

For example:

```
LC_GLOBAL
    {
    CENTRAL_HOST = "quahog"
    LOGPREFIX = "/usr/lclog"
    BACKUP_HOST = "orca"
    BACKUP_LOGPREFIX = "/usr/backuplog"
    }
```

5. If desired, make additional changes to the messaging configuration file. See Appendix G, “Configuration Files.”

6. If desired, run `show_config -c -f` *`config_file`* to check the messaging configuration file for syntax errors. For the command syntax, see Appendix B, “Commands.”

7. If desired, modify the initialization file. See Appendix F, “Initialization File.”

8. Create service entries for the UDP services if they do not exist.

   Log Central uses two UDP services for communication between the Log Central processes on the central host and Log Central processes on the managed nodes. The service entries are `lc_talk`, which has a default port of 7012, and `lc_conf`, which has a default port of 7011. These services need to be available on the central host and managed nodes. For more information, see your network administrator. Service entry examples:

   ```
   lc_conf    7011/udp
   lc_talk    7012/udp
   ```

   On Windows NT, these entries are in *`system_root`*`\drivers\etc\services` where *`system_root`* is probably similar to `C:\winnt\system32`. On UNIX, these entries are in the YP services table, which is in the `/etc/services` file.

CHAPTER

4 Configuring Multiple Instances of Log Central

The following sections describe the conditions for configuring multiple instances of Log Central and explain how to configure multiple instances:

- When To Configure Multiple Instances
- Configuring Multiple Instances

When To Configure Multiple Instances

A single instance of Log Central consists of software integrated across one central host and one or more managed notes. You need to configure multiple instances of Log Central if all the following are true:

- A set of log messages generated by a set of managed resources is independent of another set of log messages generated by another set of managed resources.
- Both sets of managed resources share at least one physical machine.
- You need to administer the sets of managed resources separately.

For example, if you have more than one BEA Tuxedo domain and you need to administer them separately, then you need a separate instance of Log Central for each BEA Tuxedo domain.

Configuring Multiple Instances

To configure multiple instances of Log Central, for each additional instance perform the following steps on the central host:

1. Define two communication services in the services database similar to `lc_conf` and `lc_talk`, using the following format:

   ```
   conf_service    port_number1/udp
   msg_service     port_number2/udp
   ```

   Make sure the domains are unused ports.

   For example:

   ```
   lc_conf_dom2    9011/udp
   lc_talk_dom2    9012/udp
   ```

2. Make a copy of the Log Central messaging configuration file. For example, *`install_dir`*`/etc/messaging.conf.dom2`, where *`install_dir`* is the directory where you installed Log Central.

3. Make a copy of the Log Central initialization file. For example, *`install_dir`*`/etc/message_processor.ini.dom2`, where *`install_dir`* is the directory where you installed Log Central.

4. Edit the `LC_GLOBAL` entry in the copied messaging configuration file.

   - Set `TALK_SERVICE` to the new `msg_service`.
   - Set `IPCKEY` to a value that is different from the value of any other instance.
   - Set `INIFILE` to the name of the copied initialization file.

   For example:

   ```
   LC_GLOBAL
       {
       CENTRAL_HOST = "quahog"
       LOGPREFIX = "/usr/lclog"
   ```

```
    TALK_SERVICE = "lc_talk_dom2"
    IPCKEY = 0xee220000
    INIFILE = "install_dir/etc/message_processor.ini.dom2"
    }
```

5. Run `lc_config` to configure the new files. For example:

   ```
   lc_config -conffile install_dir/etc/messaging.conf.dom2
   -inifile install_dir/etc/message_processor.ini.domw
   ```

   where *`install_dir`* is the directory where you installed Log Central.

6. Set environment variables.

   - Set BEA_LC_IPCKEY to the `IPCKEY` value set in step 4.
   - Set BEA_LC_CONF_SERVICE to the new `conf_service`. For example, `lc_conf_dom2`.

   For information about environment variables, see Appendix C, “Environment Variables.”

7. On each managed node, perform the previous steps using the same values.

Note: Use the `-i inifile` parameter for all database commands, such as `lc_user_list`.

CHAPTER

5 Creating Log Mappings

The following sections describe how to create log mappings:

- What Is a Log Mapping?
- Where to Define Log Mappings
- Using Multiple Log Mappings
- Mapping Options
- Date Formats for the -D Option
- Specifying Option Values
- Mapping Dates
- Working with Metacharacters
- Specifying Multiple Separators
- Working with Field Lengths
- Example: Mapping a System Log

What Is a Log Mapping?

A log mapping is a list of options. For example:

```
-S |! -o %F8 -p sony -b %F12
```

A Log Monitor uses log mappings to translate log messages into the Log Central format, which is described in Appendix A, “Log Central Message and Message Definition Formats.” Log Central provides predefined mappings for several types of log files. For descriptions of these predefined mappings, see Appendix H, “Predefined Log Mappings.” This chapter describes how to create mappings for additional types of log files.

The following table describes how the previous mapping works.

Table 5-1 Mapping Example

Mapping Option	Description
`-S |!`	Specifies that `|` and `!` are the separator characters between the fields in the incoming message
`-o %F8`	Tells the Log Monitor to use the incoming message’s 8th field as the host name in the resulting message
`-p sony`	Tells the Log Monitor to forward the message only if “sony” is in the message
`-b %F12`	Tell the Log Monitor to use the incoming message’s 12th field as the body of the resulting message

Where to Define Log Mappings

If you are going to start the Log Monitors automatically when you start Log Central with the `start_messaging` command, you need to:

- Define the log mappings in the messaging configuration file.
- Include `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file.

If you are going to start the Log Monitors manually with the `log_monitor` command, you need to define the log mappings in the message mapping file or on the `log_monitor` command line. If you use the `log_monitor` command line, you can define only one log mapping.

For information about the messaging configuration file, see Appendix G, “Configuration Files.” For information about the `start_messaging` and `log_monitor` commands, see Appendix B, “Commands.”

Using Multiple Log Mappings

When you define log mappings in a message mapping file or in the messaging configuration file, the Log Monitor tests an incoming log message against each mapping sequentially. To be forwarded to the Central Collector, a log message needs to be selected by only one of the mappings in the file. You can use multiple mappings to handle different types of log messages that come from the same managed resource.

Mapping Options

Most of the mapping options tell the Log Monitor how to map a field in an incoming message to a field in the resulting message. However, you can use the `-p` and `-x` options to discard messages. For example, if you use the `-p` option in a mapping, the

Log Monitor forwards a log message only if it contains the specified pattern. If you use the `-x` option, the Log Monitor discards log messages that contain the specified pattern. The following table describes the mapping options.

Table 5-2 Mapping Options

Option	Description
`-b` *`body`*	String that contains the body of the message.
`-D` *`date`* `%f "`*`format`*`"`	Date format to use in the resulting message. See “Date Formats for the -D Option” and “Mapping Dates.”
`-d` *`msgid`*	Message ID to use in the resulting message. The default is `1000`.
`-e` *`entity`*	Entity name to use in the resulting message. An entity name is the name that one or more Log Monitors use to register with the `proc_monitor` process. The default is `log_monitor`. Each Log Monitor on one managed node must have a unique entity name. If you run a Log Monitor as a daemon (with `-t 0`), this option is not used.
`-I` *`processID`*	Process ID to use in the resulting message. The default is the process ID of `log_monitor`.
`-M` *`log_level`*	Logging level to use in the resulting message. This is a one-character string with the following possible values: `N`—Normal `V`—Verbose `D`—Debug `S`—Special
`-m` *`subsystem`*	Subsystem name to use in the resulting message. The default is `none`.
`-n` *`function`*	Function name to use in the resulting message. The default is `none`.
`-o` *`hostname`*	Host name to use in the resulting message. The default is the machine on which the Log Monitor is running.
`-p` *`pattern`*	Tells the Log Monitor to forward a log message only if it contains the specified pattern, which can include the metacharacters described in “Working with Metacharacters.”
`-S` *`separators`*	Separators for parsing field values when the mapping includes `%F` format symbols. Table 5-4 describes `%F`. If you specify more than one separator, the Log Monitor uses all of them. For more information, see “Specifying Multiple Separators.”

Table 5-2 Mapping Options

Option	Description
`-u` *`userID`*	User ID to use in the resulting message.The default is the current user, which is the owner of the `log_monitor` process.
`-x` *`pattern`*	Tells the Log Monitor to discard a log message if it contains the specified pattern, which can include the metacharacters described in “Working with Metacharacters.” The message can still be forwarded if it satisfies another mapping in the message mapping file or messaging configuration file.

Date Formats for the -D Option

The following table describes the date formats for the `-D` option.

Table 5-3 Date Formats (%f)

Format	Description
`%%`	A literal percent sign
`%a`	Abbreviated weekday name (for example, `Sun`)
`%A`	Full weekday name (for example, `Sunday`)
`%b`	Abbreviated month name (for example, `Jan`)
`%B`	Full month name (for example, `January`)
`%d`	Day of month (1-31; leading zeroes are permitted but not required)
`%D`	Date as `%m/%d/%y`
`%h`	Same as `%b`
`%H`	Hour (0-23; leading zeroes are permitted but not required)
`%I`	Hour (0-12; leading zeroes are permitted but not required)
`%j`	Day number of year (001-366; leading zeroes are permitted but not required)

Table 5-3 Date Formats (%f)

Format	Description
`%m`	Month number (1-12; leading zeroes are permitted but not required)
`%M`	Minute (0-59; leading zeroes are permitted but not required)
`%p`	Local equivalent of AM or PM
`%r`	Time as `%I:%M:%S %p`
`%S`	Seconds (0-59; leading zeroes are permitted but not required)
`%T`	Time as `%H:%M:%S`
`%y`	Year in the century (0-99; leading zeroes are permitted but not required)
`%Y`	Year, including the century (for example, `2000`)

Specifying Option Values

An option value can be a literal or a format. You can mix literals and formats in the same option value. To use a literal, follow the option with the specific value. For example:

```
-b "This is the message body."
-m "Major function"
```

To use a format, use format symbols to extract the value from the incoming log message. For example:

```
-o %F8
```

To use a literal and a format, precede the literal with `%V`, which is a format symbol. For example:

```
-o %F11%Vmachine
```

The following table describes the format symbols.

Table 5-4 Format Symbols

Format Symbol	Description	Examples

`%C`	Starting character position for the value. Must be followed by `L` or `S` to terminate the value. `L` specifies the number of characters in the value. `S` specifies that the following character is a separator.	`-u %C10L4`
		Selects the 10th, 11th, 12th, and 13th characters from the incoming message to use as the user ID in the resulting message.
		`-u %C10S|`
		Selects the string that starts at the 10th character and ends immediately before the next `|` character to use as the user ID in the resulting message.

`%F`	Used with an integer to specify a field. Must be accompanied by the `-S` option. Fields are numbered starting with 1 (not 0).	`-m %F1 -S |` Selects the first field from the incoming message and uses it as the subsystem name in the resulting message. The field separator is the vertical bar.

`%f`	Format for the date value. This option is used with the `-D` option only. "Date Formats for the -D Option" describes the date formats.	`-D %F3%f%T`
		Selects the third field, which is in the `%T` format, to use as the date in the resulting message.
		`-D "%F1%V %F2%V %F3%f%h %d %T" -S " "`
		Selects the first field, which is in the `%h` format; the second field, which is in the `%d` format; and the third field, which is in the `%T` format, to use as the date in the resulting message. You need to use double quotes if the option is on the command line or contains one or more embedded spaces.
		There must be separators between the date format specifiers with `%f`. The same separator should appear in the value specified for `-D`. This is why `%V` is used in the first part of the specification to fill in the corresponding separators between the fields generated by Log Monitor.

`%V`	User-defined value.	`-n %C3L5%F11%V"minor function" -S :` Selects the five-character string starting at the 3rd character in the 11th field and adds the string "minor function" to use as the function name in the resulting message. The field separator is the colon.

Mapping Dates

The following rules apply for converting dates from date and time values in an incoming log message to date and time values in the Log Central log message:

- If only the weekday is in the incoming message, without specifying a day of the month, today is assumed if the incoming weekday is equal to the current weekday. Otherwise, the corresponding day from the previous six days is assumed.
- If only the month is in the incoming message, without specifying a year, the current month is assumed if the incoming month is equal to the current month. Otherwise, the corresponding month from the previous 11 months is assumed. The first day of the month is assumed if no day is provided.
- If only the time is in the incoming message, without specifying a date, today is assumed. If no hour, minute, and second are provided, the current hour, minute, and second are assumed.

The following example illustrates these rules for a current date of Tuesday September 19 12:19:47 PDT 2000.

Listing 5-1 Example for Mapping Dates

```
Input            Line in Template      Date
______________________________________________________________
Tue              %a                    Sep 19 12:19:47 PDT 2000
Mon              %a                    Sep 18 12:19:47 PDT 2000
Fri              %a                    Sep 15 12:19:47 PDT 2000
September        %B                    Sep 1 12:19:47 PDT 2000
January          %B                    Jan 1 12:19:47 PDT 2000
December         %B                    Dec 1 12:19:47 PDT 1999
Sep Tue          %b %a                 Sep 19 12:19:47 PDT 2000
Jan Sat          %b %a                 Jan 1 12:19:47 PDT 2000
Dec Tue          %b %a                 Dec 7 12:19:47 PDT 1999
Jan Fri 2003     %b %a %Y              Jan 3 12:19:47 PDT 2003
Fri 9            %a %H                 Sep 15 09:00:00 PDT 2000
Feb 10:30        %b %H:%S              Feb 1 10:00:30 PDT 2000
10:30            %H:%M                 Sep 19 10:30:00 PDT 2000
13:30            %H:%M                 Sep 19 13:30:00 PDT 2000
```

Working with Metacharacters

With the `-p` and `-x` options, you can use metacharacters to select a range of values. For example, to specify a value from 97 to 99, you can use `9[7-9]`. The following table lists the metacharacters you can use with the `-p` and `-x` options.

Table 5-5 Metacharacters

Expression	Description
`?`	Matches any single character except a newline character.
`%`	Matches the beginning of the line. For example, `%abc` matches a string only if the letters `abc` are the first three characters of a line. The `%` symbol does not have its special metacharacter role if it is not at the beginning of a line.
`$`	Matches the end of a line. For example, `xyz$` matches a string only if the letters `xyz` are the last three characters on the line. The `$` symbol does not have its special metacharacter role if it is not at the end of a line.
`@`*`c`*	Escapes the character that follows the `@`. When followed by any metacharacter, the expression matches the metacharacter itself. For example, `@%` matches a percent sign, which otherwise would be interpreted as part of an expression that starts at the beginning of a line.
`*`	Indicates zero or more occurrences of the preceding character or expression. A single character followed by an asterisk is a regular expression that matches zero or more occurrences of that one character. If the expression has multiple matches, it chooses the longest leftmost string that permits a match. For example, in a line starting `aaabaa`, the expression `a*` would match `aaa`. For another example, `[a-zA-Z][a-zA-Z]*$` matches lines that end in words. The specification is to match an entire word, that is, one containing one or more alphabetic characters.
[*`string`*]	Indicates a string. A nonempty string enclosed in square brackets matches any one character in the string. If the first character is a caret (`^`), the regular expression matches any character except a newline character and the remaining characters in the string. For example, `[^a-zA-Z0-9]` matches any nonalphanumeric character. Use a hyphen to indicate a range of consecutive ASCII characters, such as `[0-9]`.

Specifying Multiple Separators

When you use the `-S` option, you can specify multiple separator characters. For an incoming message of `abcd^xys^b|bbbb^`, Table 5-6 shows how the Log Monitor parses the message with `-S ^|`, and Table 5-7 shows how the Log Monitor parses the message with `-S ^`.

Table 5-6 Multiple Separators Example 1: -S ^|

Field Number	Contents
1	`abcd`
2	`xys`
3	`b`
4	`bbbb`

Table 5-7 Multiple Separators Example 2: -S ^

Field Number	Contents
1	`abcd`
2	`xys`
3	`b|bbbb`

The Log Monitor ignores separators at the beginning of a message. For example, the fields would be exactly the same if the incoming message was `|abcd^xys^b|bbbb^`.

Working with Field Lengths

If the length of a string-valued field in an incoming message exceeds its maximum, the Log Monitor truncates the value. For example, if an incoming message contains the user ID `Administrator`, it would be truncated to `Administ`. The following table lists the maximum field lengths.

Table 5-8 Maximum Field Lengths

Field	Maximum Length
Subsystem Name	8
User ID	8
Hostname	20
Function name	40
Entity	21
Timestamp	20
Message Body	2000

Example: Mapping a System Log

This example filters a system log. The following listing contains messages from the UNIX system log (`/var/log/syslog`).

Listing 5-2 UNIX System Log

```
May 15 11:06:02 eclipse vmunix: psig: "EM_client" signal 15 was
masked, put back.

May 16 13:51:11 eclipse lpd[8951]: /usr/spool/lpd/lpd-log: No such
file or directory

May 17 10:38:12 eclipse su: ’su webuild’ failed for emilie on
/dev/ttyp4

May 17 13:54:28 eclipse vmunix: NFS write error: on host iseult
remote file system full

May 17 13:54:37 eclipse last message repeated 13 times

May 17 14:40:42 eclipse lpd[9290]: /usr/spool/lpd/lpd-log: No such
file or directory

May 17 17:08:09 eclipse su: ’su root’ succeeded for emilie on
/dev/ttyp0
```

The following listing contains a message mapping file.

Listing 5-3 Message Mapping File

```
-M LM_VERBOSE -D "%F1%V %F2%V %F3%f%h %d %T" -S " " -m NFS -d 123
-o %F4 -p "write error" -u emilie -n %F5 -b %F6-

-D "%F1%V %F2%V %F3%f%h %d %T" -S " " -m AUTH -d 124 -o %F4 -p su:
-u emilie -n %F5 -b %F6-

-D "%F1%V %F2%V %F3%f%h %d %T" -S " " -m PRINT -d 125 -o %F4 -p lpd
-u emilie -n %F5 -b %F6-
```

The following listing shows the result of the mapping.

Listing 5-4 Result of Mapping

```
|N|May 16 13:51:11 2000|PRINT|125|eclipse|11593|emilie|lpd[8951]:
|0|1!/usr/spool/lpd/lpd-log: No such file or directory

|N|May 17 10:38:12 2000|AUTH|124|eclipse|11593|emilie|su:
|0|1!’su webuild’ failed for emilie on /dev/ttyp4

|V|May 17 13:54:28 2000|NFS|123|eclipse|11593|emilie|vmunix:
|0|1!NFS write error: on host iseult remote file system full

|N|May 17 14:40:42 2000|PRINT|125|eclipse|11593|emilie|lpd[9290]:
|0|1!/usr/spool/lpd/lpd-log: No such file or directory

|N|May 17 17:08:09 2000|AUTH|124|eclipse|11593|emilie|su:
|0|1!’su root’ succeeded for emilie on /dev/ttyp0
```

The following table explains how mapping the third log message in the UNIX system log (Listing 5-2) with the second line in the message mapping file (Listing 5-3) produces the second message in the results (Listing 5-4).

Table 5-9 Mapping Example

Mapping	Input	Meaning	Output
`-D "%F1%V %F2%V %F3%f%h %d %T" -S " "`	`May 17 10:38:12`	Time stamp: constructed from fields 1, 2, and 3.	`May 17 10:38:12 2000`
`-m AUTH`	Field not present in input.	Subsystem: specified by literal value.	`AUTH`
`-d 124`	Field not present in input.	Message ID: specified by literal value.	`124`
`-o %F4`	`eclipse`	Host name: taken from field 4.	`eclipse`
`-p su:`	`su:`	Pattern match required to forward the message.	The message can be forwarded.
`-u emilie`	Field not present in input.	User name: specified by literal value.	`emilie`

Table 5-9 Mapping Example

Mapping	Input	Meaning	Output
`-n %F5`	`su:`	Function name: taken from field 5.	`su:`
`-b %F6-`	`’su webuild’ failed for emilie on /dev/ttyp4`	Message body: constructed from field 6 to the end of line.	`’su webuild’ failed for emilie on /dev/ttyp4`
None.	None.	Reporting mode: the default, `LM_NORMAL`, which produces an `N`.	`N`
None.	None.	Process ID: the default, which is the PID of `log_monitor`.	`11593`
None.	None.	Entity: the default, which is `0`.	`0`
None.	None.	Reserved.	`1!`

CHAPTER

6 Creating Message Definitions

The following sections describe message definitions and explain how to create a message definition file:

- Overview of Message Definitions
- Creating a Message Definition File

Overview of Message Definitions

A message definition provides static information about log messages, such as recommendations for responding to errors. The Central Collector appends a message definition to each log message based on the message ID and subsystem values in the message.

Log Central provides predefined message definitions for the following types of logs:

- BEA Tuxedo logs
- BEA WebLogic Enterprise logs
- Oracle alert logs
- Windows NT event logs

This chapter describes how to create a message definition file for additional types of logs. You can also use the Log Central Console to add, modify, and delete message definitions as described in the Log Central Online Help.

A message definition file contains fields for creating one or more message definitions. The first two fields, `SUBYSTEM` and `MESSAGE_ID`, are not part of the message definition. The Central Collector uses these two fields to match a message definition to a log message. The following table describes the fields in a message definition file. If a field is in a message definition but not in succeeding definitions, the value of that field is inherited by the succeeding message definitions. In a message definition file, a string must be on one line.

Table 6-1 Fields for Creating a Message Definition

Field	Description
`SUBSYSTEM`	See “Subsystem” in Appendix A, “Log Central Message and Message Definition Formats.”
`MESSAGE_ID`	See “Message ID” in Appendix A, “Log Central Message and Message Definition Formats.”
`SUMMARY`	See “Summary” in Appendix A, “Log Central Message and Message Definition Formats.”
`SEVERITY`	See “Severity” in Appendix A, “Log Central Message and Message Definition Formats.”
`DESCRIPTION`	See “Description” in Appendix A, “Log Central Message and Message Definition Formats.”
`RECOMMENDATION`	See “Recommendation” in Appendix A, “Log Central Message and Message Definition Formats.”
`EXECUTE_ON_UPLOAD`	See “Execute on Upload” in Appendix A, “Log Central Message and Message Definition Formats.”
`TRAP_ID`	See “Trap ID” in Appendix A, “Log Central Message and Message Definition Formats.”
`TRAP_ENABLED`	See “Trap Enabled” in Appendix A, “Log Central Message and Message Definition Formats.”
`AUTO_ACKNOWLEDGE`	See “Auto Acknowledge” in Appendix A, “Log Central Message and Message Definition Formats.”

The following code is an example of a message definition.

Listing 6-1 Example of Code for Creating a Message Definition

```
{
    SUBSYSTEM         = Tuxedo
    MESSAGE_ID        = 1206
    SUMMARY           = Memory allocation failed for compression
    SEVERITY          = Critical
    DESCRIPTION       = An attempt dynamically to allocate memory from the\
operating system failed while compressing a message.
    RECOMMENDATION    = Make sure the operating system parameters are set\
correctly for the amount of memory on the machine and the amount of memory that\
can be used by a process. Reduce the memory usage on the machine or increase\
the amount of physical memory on the machine.
    EXECUTE_ON_UPLOAD = C:\bin\sendalert.exe
    TRAP_ID           = 47
    TRAP_ENABLED      = Yes
    AUTO_ACKNOWLEDGE  = Yes
}
```

Creating a Message Definition File

To create and load a message definition file:

1. Create a message definition file.

   Log Central provides a template for a message definition file in `install_dir/etc/msgdef.template`, where *`install_dir`* is the directory where you installed Log Central.

2. Run `msgdef_import` to load the message definition file.

   For the command syntax, see Appendix B, “Commands.”

CHAPTER

7 Creating Filters

The following sections describe how to define and assign filters:

- Overview of Filters
- Defining Filters
- Assigning Filters

Overview of Filters

Filters are defined and assigned in the Log Central messaging configuration file. For more information about this file, see Appendix G, “Configuration Files.” You can also use the Log Central Console to filter messages. For information about the Log Central Console, see the Log Central Online Help.

You can create filters that cause the Data Collection Agents to do the following:

- Discard (not forward) specified log messages.
- Execute a program or script when a specified log message occurs.
- Send an SNMP trap notification when a specified log message occurs.
- Save a log message to a file.

Try to use filters as little as possible. Because each log message must go through all of the filters, throughput can be adversely affected. In particular, the `COMMAND` action is very time-consuming.

To create a filter:

1. Define the filter as described in “Defining Filters.”
2. Assign the filter as described in “Assigning Filters.”

Defining Filters

The following sections explain how to define filters:

- Creating a DEFINE_FILTER Entry
- Defining a Condition
- Filtering Based on a Domain
- Defining an Action
- Suppressing an Action

Creating a DEFINE_FILTER Entry

To define a filter, create a `DEFINE_FILTER` entry in the Log Central messaging configuration file. The `DEFINE_FILTER` entry must precede the filter assignment statements in the messaging configuration file. A `DEFINE_FILTER` entry has the following syntax:

```
DEFINE_FILTER "filtername"
if condition
    {
    action_statement1
    [action_statement2]
    [action_statementN]
    }
```

filtername cannot exceed eight characters. Each action statement must be of a different type. For example, a filter cannot include two `COMMAND` action statements. For information about action statements, see “Defining an Action.”

Note: The maximum number of filters that you can define is 50. If the messaging configuration file contains more than 50 filters, the Log Central behavior becomes unpredictable.

Defining a Condition

A condition can be simple or complex. The following example shows a filter that uses a simple condition:

```
DEFINE_FILTER "DropInfo"
if (MSGID == 8)
    {
    REMOTE = "NO"
    }
```

This entry defines a filter named `DropInfo`. The filter specifies that if a message has a message ID of 8, it is dropped (not sent to the Central Collector). You can use a filter like this to drop messages that you do not want to monitor. By default, data collection agents send all messages to the Central Collector. In other words, the default value of `REMOTE` is `YES`.

To create a complex condition, use logical operators to combine simple conditions. The following table describes these logical operators.

Table 7-1 Logical Operators for Defining Conditions

Syntax	Interpretation
`!(condition)`	Evaluates to true if `condition` is false.
`(condition1) && (condition2)`	Evaluates to true if both `condition1` and `condition2` are true.
`(condition1) || (condition2)`	Evaluates to true if either `condition1` or `condition2` (or both) is true.

The following example shows a filter that drops messages that have a message ID of 8 and are from subsystem NDB:

```
DEFINE_FILTER "DropInfo"
if ((SUBSYSTEM == "NDB") && (MSGID == 8))
    {
    REMOTE = "NO"
    }
```

In a filter condition, use enough parentheses in the `if` statement to preclude ambiguities in the evaluation, because precedence rules are not followed strictly during evaluation.

You can use the message body and message header fields to define filtering conditions. You cannot use message definition fields because the Central Collector adds these fields to the message, which means that they are not available to the data collection agents. The following table describes the keywords that you can use to define conditions. String values must be enclosed in quotes.

Table 7-2 Keywords for Defining Conditions

Keyword	Data Type	Description
PID	Number	Process ID
MSGID	Number	Message ID
SUBSYSTEM	String	Subsystem
LOG_LEVEL	String	Logging level
HOST	String	Host name
FUNCTION	String	Internal function
ENTITY	String	Entity name of Log Monitor Use this keyword to perform domain-based filtering as described in “Filtering Based on a Domain.”
USER	String	User ID
MSGBODY	String	String to test for a match in the body of the message

The following table describes the relations that you can use to define conditions.

Table 7-3 Relations for Defining Conditions

Symbol	Meaning
`==`	Numeric: Is equal to String: Is identical to
`!=`	Numeric: Is not equal to String: Does not match
`>=`	Numeric: Greater than or equal to String: Contains or is the same as
`<=`	Numeric: Less than or equal to String: Is a substring of or is the same as
`>`	Numeric: Greater than String: Contains and is not the same as
`<`	Numeric: Less than String: Is a substring of and is not the same as

Filtering Based on a Domain

Domain-based filtering enables you to filter messages based on the Log Monitor entity name. For example, if a system has three managed nodes and each managed node has a Log Monitor with an entity name of LogMonWLE, these Log Monitors constitute a domain.

You can create filters for Log Monitors based on the domain. For example:

```
DEFINE_FILTER "DropInfoWLE"
if (MONITOR == "LogMonWLE" && MSGID == 8)
    {
    REMOTE = "NO"
    }
```

To implement filtering based on a domain, define the Log Monitor entity names in one of the following places:

- `-e` option on the `log_monitor` command line.
- Name of the `DEFINE_LOG_MONITOR` entry in the messaging configuration file.

For information about the `log_monitor` command, see Appendix B, “Commands.” For information about the messaging configuration file, see Appendix G, “Configuration Files.”

Defining an Action

You can specify one or more actions for a Data Collection Agent to perform when a condition is true. The following table describes the types of action statements that are possible.

Table 7-4 Types of Actions

To Perform the Following Action:	Use the Following Action Statement:
Drop a log message	`REMOTE = "NO"`
Forward a log message	`REMOTE = "YES"`
Run a script or program	`COMMAND = "`*`executable_path`*` [arguments]"` where *`executable_path`* is the full pathname for the script or program
Send an SNMP trap	`TRAPID = `*`trap_number`*
Save a log message to a file	`LOCAL = "`*`filename`*`"` where *`filename`* is the full pathname for the file.

Suppressing an Action

In addition to using a filter condition to determine whether or not to perform an action, you can also apply suppression criteria to actions. Suppression criteria are based on one or both of the following values:

- Time interval—Any duplication within a specified length of time is ignored. For example, duplicate log messages can be suppressed for 30 minutes.
- Number of occurrences—Any duplication within a specified number of occurrences is ignored. For example, only every 50th occurrence of a duplicate log message is recognized.

You can apply suppression criteria to all actions in a filter or to a selected subset of actions in a filter. The following table describes the keywords for defining suppression criteria.

Table 7-5 Keywords for Defining Suppression Criteria

Keyword	Data Type	Description
INTERVAL	Time	Time interval during which Log Central ignores duplicate log messages. The format is `*xx*h:*yy*m:*zz*s` where: ■ *xx* is the number of hours. ■ *yy* is the number of minutes. ■ *zz* is the number of seconds. For example: `01h:30m:24s`.
OCCURRENCES	Number	Number of occurrences during which Log Central ignores duplicate log messages.

To use suppression criteria:

- Enclose one or more actions in a `DO` statement.
- Close the `DO` statement with an `IGNORE_DUPLICATES_WITHIN` condition.
- Define the suppression criteria in the `IGNORE_DUPLICATES_WITHIN` condition.

For example:

```
DEFINE_FILTER "fatal"
if (SUBSYSTEM == "KERNEL" && MSGBODY >= "fatal")
{
    DO
    {
        COMMAND="/usr/mybin/page_admin"
        TRAPID=123
        REMOTE="YES"
    }IGNORE_DUPLICATES_WITHIN (INTERVAL="00h:30m:00s" || OCCURRENCES=100)
    LOCAL="/usr/local/logs/fatal"
}
```

In this example, the actions in the `DO` statement are suppressed (ignored) if the log message occurs within 30 minutes after the first duplicate log message or if the log message precedes the 100th occurrence of the same log message.

A `REMOTE` statement exhibits the following special behaviors when included in a filter that includes suppression criteria:

- A `REMOTE="YES"` statement causes only the messages allowed by the suppression criteria to be forwarded. If the filter does not include a `REMOTE="YES"` statement, then all messages are forwarded (unless the filter includes a `REMOTE="NO"` statement). The `REMOTE="YES"` statement can occur inside the `DO` statement or outside it; the effect is the same in either location.
- A `REMOTE="NO"` statement inside the `DO` statement causes one message to be forwarded; all other messages are dropped.
- A `REMOTE="NO"` statement outside the `DO` statement causes all messages to be dropped.

If a filter that includes suppression criteria does not include a `REMOTE` statement, then all messages are forwarded.

Assigning Filters

The following sections explain how to assign and turn off filters:

- Overview of Assigning Filters
- Assigning a Global Filter
- Turning Off a Global Filter
- Assigning a Local Filter

Overview of Assigning Filters

You can assign a filter globally or locally. A global filter affects the entire system. A local filter affects a managed node. The simplest approach is to use global filters for situations that apply to the greatest number of nodes, and specify any exceptions locally.

Filters are assigned or turned off in the Log Central messaging configuration file. For information about this file, see Appendix G, “Configuration Files.”

Assigning a Global Filter

To assign a global filter, create a `FILTER` statement in the `LC_GLOBAL` entry in the Log Central messaging configuration file. You can use multiple `FILTER` statements to assign multiple global filters. The syntax for the `FILTER` statement is:

```
FILTER = "filtername"
```

where *`filtername`* cannot exceed eight characters.

For example:

```
LC_GLOBAL
    {
    CENTRAL_HOST = "quahog"
    LOGPREFIX = "/usr/lclog"
    BACKUP_HOST = "orca"
    BACKUP_LOGPREFIX = "/usr/backuplog"
    FILTER = "BankTrap"
    }
```

Turning Off a Global Filter

To turn off the global filters for a particular managed node, use the `GLOBAL_FILTER = "NO"` statement in the `MANAGED_NODE` entry in the Log Central messaging configuration file. For example:

```
MANAGED_NODE
    {
    HOSTNAME = "bigiron"
    GLOBAL_FILTER = "NO"
    }
```

Assigning a Local Filter

To assign a local filter, create a `FILTER` statement in a `MANAGED_NODE` entry in the Log Central messaging configuration file. You can use multiple `FILTER` statements to assign multiple local filters to a node. The syntax for the `FILTER` statement is:

```
FILTER = "filtername"
```

`filtername` cannot exceed eight characters.

For example:

```
MANAGED_NODE
    {
    HOSTNAME = "marmalade"
    FILTER = "F2"
    FILTER = "F3"
    GLOBAL_FILTER = "NO"
    }
```

CHAPTER

8 Integrating SNMP

The following sections explain how to integrate SNMP with Log Central:

- Overview of SNMP and MIBs in Log Central
- Setting Up SNMP Management
- What Traps Are Generated?

Overview of SNMP and MIBs in Log Central

SNMP lets you monitor and manage log resources from an SNMP-compliant system manager. You can configure the following types of SNMP traps that are generated for incoming messages:

- Basic traps

  Basic traps are defined in the Basic Trap Configuration window in the Log Central Console. The Central Collector generates an SNMP trap when it receives a log message that is mapped to an SNMP trap.

- Advanced traps

  Advanced traps are defined in filters in the message configuration file. A Data Collection Agent generates an SNMP trap when it receives a message that matches the filtering criteria.

An SNMP trap consists of variables defined in a Management Information Base (MIB). A MIB describes the attributes of a managed resource in a way that an SNMP management system can understand. Log Central provides the Log Central Traps MIB, which is in the `bea_lc_trap.asn1` file. This MIB contains Log Central attributes that are used as variables in the SNMP traps. For more information about the Log Central Traps MIB, see Appendix D, "MIB Reference."

Setting Up SNMP Management

To set up Log Central for monitoring by an SNMP management system:

1. Define the destination for SNMP traps in the trap configuration file.

   The trap configuration file defines the machine and port that Log Central components use as the destination for SNMP traps. By default, this is the local host. Edit this file to point to the correct destination for the traps. For more information, see Appendix G, "Configuration Files."

2. Configure the management system to recognize the enterprise of the Log Central traps.

   The enterprise name is `beaSystemDescr` and the OID is `.1.3.6.1.4.1.140.1.1`.

3. Configure the management system to respond to Log Central traps.

   You can change the way that Log Central SNMP traps are displayed on your management console. You can also change the actions that the management system takes in response to specified events.

4. Determine which log messages that enter the Log Central system should generate SNMP traps.

5. If desired, configure basic SNMP traps.

   Use the Basic Trap Configuration window, which is in the Log Central Console, to map log messages to SNMP traps. For information about the Log Central Console, see the Log Central Online Help.

6. If desired, configure advanced SNMP traps.

   For information about creating filters to specify advanced criteria for triggering SNMP traps, see Chapter 7, "Creating Filters."

What Traps Are Generated?

The following table describes the SNMP traps that Log Central generates.

Table 8-1 Generated SNMP Traps

Type of Trap	Description	Trap Value
Basic SNMP traps	These traps are generated by the Central Collector for incoming messages	To configure these traps, use the Basic Trap Configuration window as described in the Log Central Online Help. ■ If you enable a trap based on a message definition, set the Trap Enabled field to `YES` and define the trap value in the Trap ID field. ■ If you enable a trap based on severity, define the trap value in the Trap Configuration tab in the Basic Trap Configuration window. ■ If you enable a trap based on a message definition and severity, the generated trap number is the one defined in the message definition Trap ID field even if the trap is generated because of severity.
Advanced SNMP traps	These traps are generated by the Data Collection Agents for incoming messages	To configure these traps, create filters as described in Chapter 7, “Creating Filters.” Define the trap value in the `TRAPID` action statement.

A Log Central Message and Message Definition Formats

A Log Central message consists of a message header and a message body. A message definition is associated with a message based on the message ID and subsystem values in the message header. The following sections provide detailed descriptions of the Log Central message and message definition formats:

- Example of a Log Central Message and Message Definition
- Message Header
  - Log ID
  - Logging Level
  - Date and Time
  - Subsystem
  - Message ID
  - Host
  - Process ID
  - User ID
  - Function
  - Entity
- Message Body
- Message Definition
  - Summary
  - Severity
  - Description

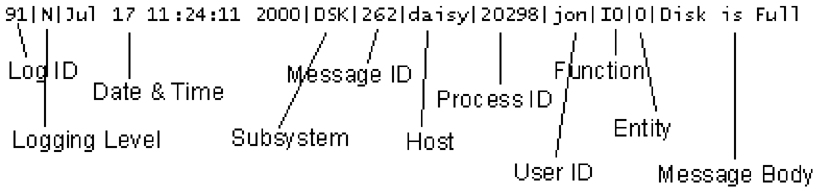
- Recommendation
- Execute on Upload
- Trap ID
- Trap Enabled
- Auto Acknowledge

For information about how Log Central constructs a log message, see “Construction of Log Messages” in Chapter 1, “Overview.”

Example of a Log Central Message and Message Definition

The following figure shows the fields that are in a Log Central message.

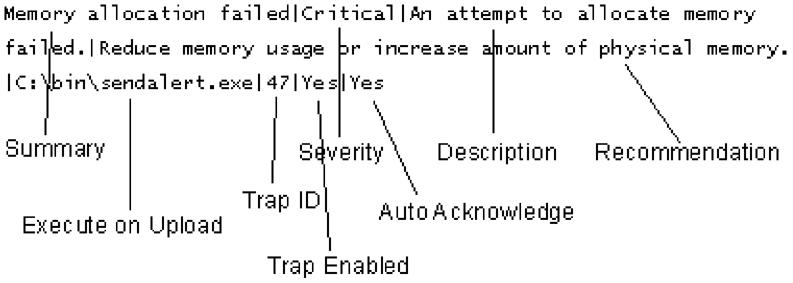
Figure A-1 Log Central Message Fields



```
91|N|Jul 17 11:24:11 2000|DSK|262|daisy|20298|jon|IO|0|Disk is Full
```

The following figure shows the fields that are in a Log Central message definition.

Figure A-2   Log Central Message Definition Fields



```
Memory allocation failed|Critical|An attempt to allocate memory
failed.|Reduce memory usage or increase amount of physical memory.
|C:\bin\sendalert.exe|47|Yes|Yes
```

Message Header

The following sections describe the fields in a message header:

- Log ID
- Logging Level
- Date and Time
- Subsystem
- Message ID
- Host
- Process ID
- User ID
- Function
- Entity

For more information about message headers, see “IL_SM_LOG_TABLE: Message Header Definition” in Appendix E, “Database Schema.”

Log ID

Description: Value that distinguishes one set of log messages from another, which lets logically separate log information be stored in one database. For example, the log IDs `IRX` and `DCS` might distinguish messages logged by a drug claim system from those logged by a document control system.

Format: String

Size: Up to 3 characters

Optional?: Optional

Corresponding Field: In the IL_SM_LOG_TABLE database table: MSG_LOG_ID

Logging Level

Description: Logging level

Format: Character

Size: 1 character

Range: Possible values:

- `N` = Normal message
- `D` = Debug message
- `V` = Verbose message
- `S` = Special message

Optional?: Not optional

Corresponding Field: In the IL_SM_LOG_TABLE database table: MSG_REP_MODE

Date and Time

Description:	Month, day, hour, minute, second, and year on the machine that generated the message.
Format:	Mmm dd hh:mm:ss yyyy
Optional?:	Not optional
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_KEY_TS

Subsystem

Description:	Subsystem that generated the message. The `subsystem_create` and `subsystem_delete` commands, which are described in Appendix B, “Commands,” let you add and delete subsystems.
Format:	String
Size:	Up to 8 characters
Optional?:	Not optional. In a message definition file, this field must be in the first definition, and is optional for the succeeding definitions.
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_SS_NAME

Message ID

Description: Identifies the type of message within a subsystem. Together, the Subsystem and Message ID uniquely identify a message.

Format: Integer

Range: `1` through `99999`

The convention for message ID values is:

- `1` through `49999` = Activity and error messages defined by application developers
- `50000` through `89999` = Reserved for future use
- `90000` = Startup message
- `90001` = Shutdown message
- `90002` = Transaction processing message
- `90100` = Transaction processing message (NCPDP)
- `90201` through `99999` = Reserved for future use

Optional?: Not optional

Corresponding Field: In the IL_SM_LOG_TABLE database table: MSG_ID

Host

Description: Network host for the system that generated the message. This value can include dots, underscores, and dashes.

Format: String

Size: Up to 64 characters

Optional?:	Not optional
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_HOST_NAME

Process ID

Description:	PID of the process that generated the message.
Format:	Integer
Range:	`1` through `99999`
Optional?:	Not optional
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_PID

User ID

Description:	User ID of the process that generated the message.
Format:	String
Size:	Up to 8 characters
Optional?:	Not optional
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_UID

Function

Description: Internal function that generated the message. The convention for function values is:

- Function name if logged by a library function
- Process name if logged by a main process function

Format: String

Size: Up to 40 characters

Optional?: Optional

Corresponding Field: In the IL_SM_LOG_TABLE database table: MSG_FCT_NAME

Entity

Description: Value that helps correlate a message with messages from other managed nodes. You can use this value to define a domain. The domain can be a Tuxedo domain or WebLogic Enterprise domain, but is not required to be one of these types of domains. The default value is “None.”

For example, if a system has three managed nodes and each managed node has a Log Monitor with an entity name of LogMonWLE, these Log Monitors constitute a domain.

For another example, if you have a Tuxedo domain for sales information, you can set the entity name to “Sales” for each Log Monitor in the domain.

For information about setting and using the entity value, see Chapter 7, “Creating Filters,” and the section “DEFINE_LOG_MONITOR” in Appendix G, “Configuration Files.”

Format: String

Size: Up to 21 characters

Optional?: Optional

Corresponding Field: In the IL_SM_LOG_TABLE database table: MSG_TRAN_KEY

Message Body

Description: Text determined by the application developer.

Format: String

Size:	Up to 2000 characters
Optional?:	Not optional
Corresponding Field:	In the IL_SM_LOG_TABLE database table: MSG_TEXT

Message Definition

The following sections describe the fields in a message definition:

- Summary
- Severity
- Description
- Recommendation
- Execute on Upload
- Trap ID
- Trap Enabled
- Auto Acknowledge

For more information about message definitions, see:

- Chapter 6, “Creating Message Definitions.”
- “IL_MSG: Message Definition” in Appendix E, “Database Schema.”
- “Changing Message Definitions,” in the Log Central Online Help.

Summary

Description: Summary of the “Description” field which describes the condition or event that the message is reporting.

Format: String

Size: Up to 40 characters

Optional?: Not optional

Corresponding Fields: In a message definition file: `SUMMARY`

In the IL_MSG database table: LOG_SDESC

Severity

Description: Possible values:

- `Informational` = Activity message.
- `Warning` = Problem that does not need immediate attention.
- `Minor` = Performance or service degradation may result.
- `Major` = Problem that needs immediate attention, or the system performance may deteriorate.
- `Critical` = Problem that need immediate attention, or the system may crash.

Format: String

Size: N/A

Optional?:	Not optional. In a message definition file, this field must be in the first definition, and is optional for the succeeding definitions.
Corresponding Fields:	In a message definition file: `SEVERITY` In the IL_MSG database table: LOG_MSG_SEVERITY

Description

Description:	Description of the condition or event that the message is reporting.
Format:	String
Size:	With the Recommendation field, up to 2 Gb.
Optional?:	Optional
Corresponding Fields:	In a message definition file: `DESCRIPTION` In the IL_MSG database table: LOG_MSG_DESC_REC

Recommendation

Description:	Recommended action for handling the message.
Format:	String
Size:	With the Description field, up to 2 Gb.
Optional?:	Optional
Corresponding Fields:	In a message definition file: `RECOMMENDATAION` In the IL_MSG database table: LOG_MSG_DESC_REC

Execute on Upload

Description:	Path to a script or executable file that the Central Collector runs when it stores the message in the Log Central database.
Format:	String
Size:	Up to 40 characters
Optional?:	Optional
Corresponding Fields:	In a message definition file: `EXECUTE_ON_UPLOAD` In the IL_MSG database table: LOG_MSG_PARSE_FNC

Trap ID

Description:	Basic SNMP trap to generate for the message. For information about generating basic SNMP traps, see “Configuring SNMP Traps” in the Log Central Online Help.
Format:	Integer
Range:	`1` through `99999`
Optional?:	Optional
Corresponding Fields:	In a message definition file: `TRAP_ID` In the IL_MSG database table: LOG_MSG_TRAP_ID

Trap Enabled

Description: Flag that indicates whether or not to generate a basic SNMP trap for the message. For information about generating basic SNMP traps, see “Configuring SNMP Traps” in the Log Central Online Help.

Format: String

Range: `YES`

`NO` (default)

Optional?: Optional

Corresponding Fields: In a message definition file: `TRAP_ENABLED`

In the IL_MSG database table: LOG_MSG_TRAP_ENABLED

Auto Acknowledge

Description: Flag that indicates whether or not to automatically acknowledge the message. One reason to acknowledge messages is to keep track of the system problems that are currently being resolved or actively investigated.

Format: String

Range: `YES`

`NO` (default)

Optional?: Optional

Corresponding Fields: In a message definition file: `AUTO_ACKNOWLEDGE`

In the IL_MSG database table: LOG_MSG_AUTO_ACK

B Commands

The following table and subsequent sections describe the Log Central commands.

Table B-1 Commands

Command	Description
`lc_config`	Runs the Host Configuration utility
`lc_create_schema`	Creates a database schema
`lc_drop_schema`	Drops a database schema
`lc_launch`	Starts the Log Central Console
`lc_user_create`	Creates a user in the database
`lc_user_delete`	Deletes a user from the database
`lc_user_modify`	Changes a user’s password in the database
`lc_user_list`	Lists all the users in the database
`log_monitor`	Starts a Log Monitor
`msgdef_delete`	Deletes message definitions from the database
`msgdef_export`	Copies message definitions from the database to a text file
`msgdef_import`	Loads a message definition file into the database
`msg_reader`	Displays the current intermediate log file
`msg_test`	Generates test messages
`show_config`	Checks the syntax of the messaging configuration file or displays information about Log Central

Table B-1 Commands

Command	Description
`start_messaging`	Starts the Log Central processes
`stop_messaging`	Stops the Log Central processes
`subsystem_create`	Creates a subsystem in the database
`subsystem_delete`	Deletes a subsystem from the database

lc_config

Summary: Runs the Host Configuration utility. Run this command before starting Log Central.

Syntax: `lc_config [-inifile` *`initialization_file`*`] [-conffile` *`configuration_file`*`] [-fn` *`fontname`*`] [-fs` *`fontsize`*`] [-h]`

Options and Arguments:

`-inifile` *`initialization_file`*

Path for the initialization file to create. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. If you use a nondefault value, you need to set `INIFILE` in the messaging configuration file. For a description of the initialization file, see Appendix F, "Initialization File." For a description of the messaging configuration file, see Appendix G, "Configuration Files."

`-conffile` *`configuration_file`*

Path for the messaging configuration file to create. The default is *`install_dir`*`/etc/messaging.conf`, where *`install_dir`* is the directory where you installed Log Central. If you use a nondefault location, set the BEA_LC_HOST_CONF environment variable. For a description of the messaging configuration file, see Appendix G, "Configuration Files." For a description of BEA_LC_HOST_CONF, see Appendix C, "Environment Variables."

`-fn` *`fontname`*

Name of the font to use in the Host Configuration utility.

`-fs` *`fontsize`*

Size of the font to use in the Host Configuration utility.

`-h`

Displays help information about the command.

Description:

This command runs the Log Central Host Configuration utility, which creates the initialization file and the messaging configuration file. For more information, see Chapter 3, "Configuring the Central Host."

lc_create_schema

Summary: Creates a database schema. Run this command before starting Log Central.

Syntax: `lc_create_schema [-inifile inifilename] [-h]`

Options and Arguments:

`-inifile inifilename`

Path for the initialization file. The default is `install_dir/etc/msg_processor.ini`, where `install_dir` is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command creates the database table definitions that Log Central uses to store messages and message definitions. For a description of these tables, see Appendix E, “Database Schema.”

If an error occurs or if `lc_create_schema` aborts, run `lc_drop_schema` to clean up files that may have been created, try to correct the problem, and recreate the schema.

lc_drop_schema

Summary: Drops a database schema. Run this command before starting Log Central.

Syntax: `lc_drop_schema [-inifile inifilename] [-h]`

Options and Arguments:

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description:

This command drops the Log Central database schema.

If `lc_create_schema` aborts, you need to call `lc_drop_schema` before calling `lc_create_schema` again. Ignore all error messages that `lc_drop_schema` generates because it may try to drop tables and synonyms that were not created. You might also run `lc_drop_schema` before reinstalling Log Central.

For a description of the database schema, see Appendix E, “Database Schema.”.

lc_launch

Summary:

Starts the Log Central Console.

Syntax:

`lc_launch [-p` *`port`*`] [-h` *`central_host`*`] [-n` *`msgs`*`] [-fn` *`fontname`*`] [-fs` *`fontsize`*`] [-b` *`browser`*`] [-h]`

Options and Arguments:

`-p` *port*
: Message Processor’s port number. The default is 7001.

`-h` *central_host*
: Name of the machine that the Central Collector is running on. If you do not provide this value, the machine on which you are running this command is considered to be the central host.

`-n` *msgs*
: Maximum number of messages or message definitions to display. The default is `1000`.

`-fn` *fontname*
: Name of the font for displaying text in the Log Central Console. The default is Times Roman.

`-fs` *fontsize*
: Size of the font for displaying text in the Log Central Console. The default is 12 points.

`-b` *browser*
: Web browser for displaying the Log Central Online Help. Use the name of an executable that is available in the current PATH, or provide a complete pathname. The default is Netscape Navigator.

`-h`
: Displays help information about the command.

Description: This command starts the Log Central Console. For information about the Console, see the Log Central Online Help.

lc_user_create

Summary: Creates a user in the database. You can run this command while Log Central is running.

Syntax: `lc_user_create -u` *username* `-p` *password* `[-inifile` *inifilename*`] [-h]`

Options and Arguments:

`-u` *`username`*

Name of the user to create. This value can be up to 10 characters and can contain any alphanumeric character, including a hyphen or underscore.

`-p` *`password`*

User password. This value can be up to 10 characters and can contain any printable ASCII character.

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command creates a new user in the Log Central database.

lc_user_delete

Summary: Deletes a user from the database. You can run this command while Log Central is running.

Syntax:

```
lc_user_delete -u username -p password [-inifile
inifilename] [-h]
```

Options and Arguments:

`-u` *`username`*

Name of the user to delete.

`-p` *`password`*

Password of the user to delete.

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, "Initialization File."

`-h`

Displays help information about the command.

Description: This command deleted a user from the Log Central database.

lc_user_modify

Summary: Changes a user's password in the database. You can run this command while Log Central is running.

Syntax:

```
lc_user_modify -u username -p oldpassword -n
newpassword [-inifile inifilename] [-h]
```

Options and Arguments:

`-u` *`username`*

Name of the user.

`-p` *`oldpassword`*

User’s current password.

`-n` *`newpassword`*

User’s new password. This value can be up to 10 characters and can contain any printable ASCII character.

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command changes a user password in the Log Central database.

lc_user_list

Summary: Lists all the users in the database. You can run this command while Log Central is running.

Syntax: `lc_user_list [-inifile` *`inifilename`*`] [-h]`

Options and Arguments:

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command lists all the users in the Log Central database.

log_monitor

Summary: Starts a Log Monitor or forwards log files.

Syntax: To start a Log Monitor with predefined mappings:

```
log_monitor -i filename -P
predefined_mapping [-t time] [-p pattern] [-x
pattern] [-e entityname] [-h]
```

To start a Log Monitor with mappings in a message mapping file:

```
log_monitor -i filename -f mapping_filename
[-t time] [-c] [-e entityname] [-h]
```

To start a Log Monitor with mapping specified on the command line and optional predefined mapping:

```
log_monitor -i filename [-P
predefined_mapping] [-t time] [-e
entityname] [-b body] [-D date_and_format]
[-d msgid] [-I processID] [-M log_level] [-m
subsystem] [-n function] [-o hostname] [-p
pattern] [-S separators] [-u userID] [-x
pattern] [-h]
```

To forward log files:

```
log_monitor -i filename -P
predefined_mapping -e entityname [-h]
```

Options and Arguments:

`-i` *`filename`*

Incoming log file to use. You can use a hyphen to specify standard input:

```
-i -
```

To forward log files, set *`filename`* to *`pathname/file*`* where *`pathname`* is the directory that contains the log files and *`file`* specifies the characters that are the same in each log file name. For example, *`filename`* could be:

```
C:\tuxedo\logfiles\ULOG*
```

`-P` *`predefined_mapping`*

Tells the Log Monitor to use predefined mapping. Possible values:

`LM`—Log Central temporary log files. For more information, see “Using the LM Predefined Mapping” in the Description section.

`NTEVENT`—Windows NT event log

`ORACLE`—Oracle alert logs

`TUXEDO`—BEA Tuxedo user logs

`-t` *`time`*

Length of time to wait between forwarding each log message. The default is one second. Use `0` to forward once and then stop. Using `0` causes the Log Monitor to run as a daemon.

`-e` *`entityname`*

Entity name that the Log Monitor uses to register with the `proc_monitor` process. The default is `log_monitor`. Each Log Monitor on one managed node must have a unique entity name. If you run a Log Monitor as a daemon (with `-t 0`), this option is not used.

`-f` *`mapping_filename`*

Message mapping file that contains the mappings.

`-c`

Applies the mappings in the message mapping file to a log message only up to the first match.

`-h`

Displays help information about the command.

For descriptions of the remaining options, see Table 5-2 in Chapter 5, “Creating Log Mappings.”

Description: This command starts a Log Monitor manually. To start log monitors automatically, use the `start_messaging` command. For information about mapping, see Chapter 5, “Creating Log Mappings.”

Using the LM Predefined Mapping

Log Central creates temporary log files. There are two abnormal situations where you might need to use Log Monitor to recover the contents of these files:

- A Message Sender writes log messages to a temporary file if the Central Collector is unavailable. When the Central Collector becomes available, the Message Sender automatically forwards the log messages from the temporary file to the Central Collector. If the Message Sender dies before it can recover the file, the temporary file may not be recovered automatically. If this happens, you can use the LM predefined mapping to recover the contents of the temporary file.
- The Message Receiver writes incoming messages to an intermediate file, and the Message Processor reads this file to insert log messages into the database. If the Message Processor does not process an intermediate file, you can use the LM predefined mapping to recover the contents of the intermediate file.

msgdef_delete

Summary: Deletes message definitions from the database. Run this command before starting Log Central.

Syntax: `msgdef_delete [-f` *`filename`*`] [-inifile` *`inifilename`*`] [-h]`

Options and Arguments:

`-f` *`filename`*

Path for the file that contains the message definitions to delete. Only the subsystem names and message IDs are necessary. If you do not provide a file name, `msgdef_delete` accepts text from the standard terminal input. For a description of the message definition file, see Chapter 6, “Creating Message Definitions.”

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command deletes message definitions from the Log Central database. You can also use the `subsystem_delete` command to delete message definitions.

msgdef_export

Summary: Copies message definitions from the database to a text file. You can run this command while Log Central is running.

Syntax:

```
msgdef_export [-f filename] [-s subsystem_name
[subsystem_name]...] [-inifile inifilename] [-h]
```

Options and Arguments:

`-f filename`

Path for the file that contains the message definitions to delete. Only the subsystem names and message IDs are necessary. If you do not provide a file name, `msgdef_export` accepts text from the standard terminal input. For a description of the message definition file, see Chapter 6, “Creating Message Definitions.”

`-s subsystem_name`

Name of the subsystem for which to copy the message definitions. You can provide multiple *`subsystem_name`* values to copy message definitions for multiple subsystems. If you do not provide a subsystem name, `msgdef_export` copies the message definitions for all the subsystems.

`-inifile inifilename`

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command copies message definitions from the Log Central database to a text file.

msgdef_import

Summary: Loads a message definition file into the database. You can run this command while Log Central is running.

Syntax: `msgdef_import [-f *filename*] [-inifile *inifilename*] [-h]`

Options and Arguments:

`-f` *`filename`*

Path for the file that contains the message definitions to delete. If you do not provide a file name, `msgdef_export` accepts text from the standard terminal input. For a description of the message definition file, see Chapter 6, “Creating Message Definitions.”

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command loads a message definition file into the Log Central database.

msg_reader

Summary: Displays the current intermediate log file.

Syntax: `msg_reader [-e] [-n] *pathname* [-h]`

Options and Arguments:

`-e`

Starts writing from the end of the intermediate file.

`-n`

Formats the logged messages before writing them. Do not use this option if you run `msg_reader` with `log_monitor`.

`pathname`

Pathname of the file to open for reading. The command adds the `.cur` extension to the pathname before opening the file.

`-h`

Displays help information about the command.

Description: This command continuously reads the current intermediate file that the Message Receiver constructs, and writes the file contents to the standard output.

msg_test

Summary: Generates test messages.

Syntax:

```
msg_test [-i] [-l length] [-n messages] [-s
subsystem] [subsystem]...] [-t interval] [-h]
```

Options and Arguments:

`-i`
: Invokes the interactive mode. The `-i` option overrides the `-t` option.

`-l` *`length`*
: Message body length. The default is 40.

`-n` *`messages`*
: Number of messages to generate. The default is `1`.

`-s` *`subsystem`*
: Subsystem name to use in the message header. You can specify multiple subsystems. The default is LC.

`-t` *`interval`*
: Time interval, in seconds, to wait between messages. The default is 0.

`-h`
: Displays help information about the command.

Description: This command tests the flow of messages. You can also use this command to print performance data. Use the Message Browser in the Log Central Console to see the test results.

For information about the Log Central Console, see the Log Central Online Help.

show_config

Summary: Checks the syntax of the messaging configuration file or displays information about Log Central.

Syntax:

```
show_config -c [-f config_file] | -g | -p | -d | -n
entity_name [-h]
```

Options and Arguments:

`-c`

Checks the messaging configuration file for syntax errors.

`-f` *`config_file`*

Name of the messaging configuration file. If you do not specify a name, `show_config` uses `messaging.conf`.

`-g`

Displays the Log Central shared memory information to `stdout`.

`-p`

Displays the Process Monitor shared memory information to `stdout`.

`-d`

Displays the Log Central and Process Monitor shared memory information to `stdout`. This option is the same as using `-g` and `-p`.

`-n` *`entityname`*

Displays detailed information about the specified entity (process) to `stdout`. Possible values for *`entityname`*:

```
log_monitor
msg_processor
msg_receiver
msg_sender
proc_monitor
start_messaging
```

`-h`

Displays help information about the command.

Description:

This command checks the syntax of the messaging configuration file or displays information about Log Central. For a description of the messaging configuration file, see Appendix G, “Configuration Files.”

start_messaging

Summary: Starts the Log Central processes.

Syntax:

```
start_messaging [-f config_file] [-q] [-v]
[central_host] [backup_central_host] [-h]
```

Options and Arguments:

`-f` *`config_file`*

Name of the Log Central messaging configuration file to use instead of `messaging.conf`. This option is available only for Central Collectors.

`-q`

Makes the process quiet, which means that no informational messages are displayed.

`-v`

Displays informational and debug messages.

`central_host`

Host name of the machine for the primary Central Collector. If you do not provide this name, the machine on which you are running this command is considered to be the central host. You must provide this name when you run `start_messaging` on a backup host or a managed node.

`backup_central_host`

Host name of the machine for the secondary Central Collector. You do not need to provide this value for the central host unless you are running a Data Collection Agent on the central host, in addition to the primary Central Collector.

`-h`

Displays help information about the command.

Description: This command starts the Log Central processes for a Central Collector or a Data Collection Agent.

For a Central Collector, `start_messaging` starts the following processes:

- Message Receiver (`msg_receiver`)
- Message Processor (`msg_processor`)
- Process Monitor (`proc_monitor`)
- Message Sender (`msg_sender`)
- If you included `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file, then `start_messaging` also starts the specified Log Monitors.

For a Data Collection Agent, `start_messaging` starts the following processes:

- Message Sender (`msg_sender`)
- Process Monitor (`proc_monitor`)
- If you included `LOG_MONITOR` statements in the `MANAGED_NODE` entry in the messaging configuration file, then `start_messaging` also starts the specified Log Monitors.

The `start_messaging` process on a managed node uses the UDP service, which is defined by the BEA_LC_CONF_SERVICE environment variable, to connect to the `start_messaging` process on the central host. Then the `start_messaging` process on the central host downloads the local host’s messaging configuration. If BEA_LC_CONF_SERVICE is not defined, `start_messaging` uses the `lc_conf` service.

The Process Monitor monitors the `start_messaging` process to make sure that it continues to run and restarts it if it dies. The `start_messaging` process monitors the `proc_monitor` process and restarts it if it dies.

stop_messaging

Summary: Stops the Log Central processes.

Syntax: `stop_messaging [-q] [-v] [-h]`

Options and Arguments:

`-q`

Makes the process quiet, which means that no informational messages are displayed.

`-v`

Displays informational and debug messages.

`-h`

Displays help information about the command.

Description: This command stops the Log Central processes.

subsystem_create

Summary: Creates a subsystem in the database. You can run this command while Log Central is running.

Syntax:

```
subsystem_create -s subsystem_name [subsystem_name]
-d subsystem_description [-inifile inifilename]
[-h]
```

Options and Arguments:

`-s` *`subsystem_name`*

Name of a subsystem. This value can be up to eight characters, must be entirely in upper case, and must be unique in the Log Central database. You can provide multiple `subsystem_name` values to create multiple subsystems.

`-d` *`subsystem_description`*

Short description of the subsystem. This value can be up to 40 characters. If it contains more than one word, enclose the entire value in double quotes.

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description: This command creates a subsystem in the Log Central database. You might want to partition the Log Central database, separating various messages (such as NT events, Oracle messages, BEA Tuxedo messages, and so on) into different categories. Each category represents a resource that generates messages. These resources are called subsystems. The subsystem is one of the unique attributes in a log message.

subsystem_delete

Summary: Deletes a subsystem from the database. You can run this command while Log Central is running.

Syntax:

```
subsystem_delete -s subsystem_name [subsystem_name]
[-inifile inifilename] [-h]
```

Options and Arguments:

`-s` *`subsystem_name`*

Name of a subsystem. You can provides multiple *`subsystem_name`* values to delete multiple subsystems.

`-inifile` *`inifilename`*

Path for the initialization file. The default is *`install_dir`*`/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. For a description of the initialization file, see Appendix F, “Initialization File.”

`-h`

Displays help information about the command.

Description:

This command deletes a subsystem from the Log Central database. The command deletes all the message definitions for the subsystem, but does not delete the messages.

APPENDIX

C Environment Variables

The following sections describe Log Central environment variables and how to set them:

- Setting an Environment Variable
- BEA_LC_CONF_SERVICE
- BEA_LC_HOST_CONF
- BEA_LC_IPCKEY
- BEA_LC_MONITOR_WAKEUP_INTVL
- BEA_LC_PROC_MAX_RESTARTS
- BEA_LC_PROC_RESTART_INTVL
- BEA_LC_TALK_SERVICE
- BEA_LC_TRAP_CONF

Setting an Environment Variable

To set an environment variable, use the `setenv` command on UNIX or the `SET` command on Windows NT at a DOS prompt. The following table provides examples.

Table C-1 Examples: Setting an Environment Variable

Platform	Example
C shell on UNIX	`setenv BEA_LC_TRAP_CONF /usr/home/myconfig.conf`
Windows NT	`SET BEA_LC_TRAP_CONF C:\usr\home\myconfig.conf`

BEA_LC_CONF_SERVICE

This environment variable defines the name of the User Datagram Protocol (UDP) service that the `start_messaging` processes on the central host and managed nodes use to communicate with each other. The default is `lc_conf`. For more information about the UDP service, see Chapter 3, "Configuring the Central Host," and Chapter 4, "Configuring Multiple Instances of Log Central."

BEA_LC_HOST_CONF

This environment variable defines the location of the Log Central messaging configuration file. The default is *`install_dir`*`/etc/messaging.conf`, where *`install_dir`* is the directory where you installed Log Central. For a description of the messaging configuration file, see Appendix G, "Configuration Files."

BEA_LC_IPCKEY

This environment variable defines the interprocess communication (IPC) key that Log Central uses, and also acts as an identifier for Log Central. If you are running multiple Log Central systems, each system needs a different value for its IPC key. BEA_LC_IPCKEY must be the same as `IPCKEY` in the `LC_GLOBAL` entry in the Log Central messaging configuration file. The default is 0xeeee0000.

The `start_messaging` process on the central host reads `IPCKEY` from the Log Central messaging configuration file and passes it to the processes that it starts (`ipc_config`, `proc_monitor`, `msg_sender`, `msg_receiver`, `msg_processor`). Log Central processes that are not started by `start_messaging` on the central host, such as `log_monitor`, `msg_test`, and `start_messaging` on the managed nodes, read BEA_LC_IPCKEY.

BEA_LC_MONITOR_WAKEUP_INTVL

This environment variable defines the time interval, in milliseconds, at which the Process Monitor checks all registered processes. The default is `5000` (5 seconds).

BEA_LC_PROC_MAX_RESTARTS

This environment variable defines the maximum number of restarts that a process can make within the time specified by BEA_LC_PROC_RESTART_INTVL. The default is `4`.

BEA_LC_PROC_RESTART_INTVL

This environment variable defines the length of time, in seconds, during which the Process Monitor can restart a process. The maximum number of restarts during this interval is defined by BEA_LC_PROC_MAX_RESTARTS. The default is `600`.

BEA_LC_TALK_SERVICE

This environment variable defines the service name that the data collection agents use when connecting to the Central Collector. The default is `lc_talk`.

BEA_LC_TRAP_CONF

This environment variable defines the location of the trap configuration file, which defines the location of SNMP management stations that are configured to receive SNMP traps from Log Central. On UNIX, the default location is `/etc/lc_trap.conf`. On Windows NT, the default location is `C:\etc\lc_trap.conf`. For information about SNMP management, see Chapter 8, “Integrating SNMP.” For information about the trap configuration file, see Appendix G, “Configuration Files.”

D MIB Reference

The following sections describe the Log Central MIB:

- Overview of MIBs
- beaTrap: Log Central Traps MIB

Overview of MIBs

A Management Information Base (MIB) describes the attributes of a managed resource in a way that an SNMP management system can understand. An SNMP MIB must be written in Abstract Notation One (ASN.1) and formatted in conformity with the SNMP standards. Log Central provides a MIB that contains information for managing log resources. This file fully conforms to the SNMP standard and is ready for loading into your SNMP manager.

Note: SNMP terminology uses the term “object” while managed resource terminology uses the term “attribute.”

beaTrap: Log Central Traps MIB

The Log Central Traps MIB, `beaTrap`, is in the `bea_lc_trap.asn1` file. This MIB consists of objects that are used as variables in SNMP traps. Each SNMP trap consists of a header and a body. The body is called a variable binding. The objects in this MIB define the values that are sent in the variable binding of SNMP traps that Log Central generates. The following table lists the OID for each object in `beaTrap`. The object descriptions follow the table.

Table D-1 Objects in beaTrap

Object Name	OID
beaTrapLcLogLevel	.1.3.6.1.4.1.140.21.1
beaTrapLcTimestamp	.1.3.6.1.4.1.140.21.2
beaTrapLcSubsys	.1.3.6.1.4.1.140.21.3
beaTrapLcMid	.1.3.6.1.4.1.140.21.4
beaTrapLcHost	.1.3.6.1.4.1.140.21.5
beaTrapLcPid	.1.3.6.1.4.1.140.21.6
beaTrapLcUid	.1.3.6.1.4.1.140.21.7
beaTrapLcFunction	.1.3.6.1.4.1.140.21.8
beaTrapLcTxKey	.1.3.6.1.4.1.140.21.9
beaTrapLcVersion	.1.3.6.1.4.1.140.21.10
beaTrapLcSeverity	.1.3.6.1.4.1.140.21.11
beaTrapLcMessageBody	.1.3.6.1.4.1.140.21.12

beaTrapLcLogLevel

Description: Log level of the message for which the SNMP trap was generated.

Format: Integer

Possible Values:

- 68—Debug
- 78—Normal
- 83—Special
- 86—Verbose

Access: Read-only

beaTrapLcTimestamp

Description: Generation time of the message for which the SNMP trap was generated.

Format: Octet string

Access: Read-only

beaTrapLcSubsys

Description: Originating subsystem of the message for which the SNMP trap was generated.

Format: Octet string

Access: Read-only

beaTrapLcMid

Description:	Message ID of the message for which the SNMP trap was generated.
Format:	Integer
Access:	Read-only

beaTrapLcHost

Description:	Host name of the message for which the SNMP trap was generated.
Format:	Octet string
Access:	Read-only

beaTrapLcPid

Description:	Process ID of the message for which the SNMP trap was generated.
Format:	Integer
Access:	Read-only

beaTrapLcUid

Description: User ID responsible for the message for which the SNMP trap was generated.

Format: Octet string

Access: Read-only

beaTrapLcFunction

Description: Function name that generated the message for which the SNMP trap was generated.

Format: Octet string

Access: Read-only

beaTrapLcTxKey

Description: Entity value of the message for which the SNMP trap was generated.

Format: Octet string

Access: Read-only

beaTrapLcVersion

Description:	Unused
Format:	Integer
Access:	Read-only

beaTrapLcSeverity

Description:	Severity of the message for which the SNMP trap was generated. This value is available only for basic SNMP traps.
Format:	Integer
Possible Values:	■ `1`—Informational ■ `2`—Warning ■ `3`—Minor ■ `4`—Major ■ `5`—Critical ■ `100`—Unknown
Access:	Read-only

beaTrapLcMessageBody

Description:	Body of the message for which the SNMP trap was generated.
Format:	Octet string
Access:	Read-only

E Database Schema

The following sections describe the database table definitions that Log Central uses to store messages and message definitions.This information can help you determine which database tables to back up and when to back them up.

- IL_MSG: Message Definition
- IL_REP_MODE: Logging Level Definitions
- IL_SEV: Severity Level Definitions
- IL_SUBS: Subsystem Definitions
- IL_SM_LOG_TABLE: Message Header Definition
- IL_TRAP_CLASS: Trap Classes
- IL_USER: User Data

IL_MSG: Message Definition

The following table describes the fields in the IL_MSG database table. For more information about message definitions, see:

- Chapter 6, “Creating Message Definitions.”
- “Message Definition” in Appendix A, “Log Central Message and Message Definition Formats.”
- “Changing Message Definitions,” in the Log Central Online Help.

Table E-1 IL_MSG

Field	Null?	Type	Size	Description
LOG_SUBS_NAME	Not null	CHAR	8	Subsystem. See “IL_SUBS: Subsystem Definitions.”
LOG_MSG_ID	Not null	NUMBER	5	Message ID.
LOG_SDESC	Not null	CHAR	40	Summary of the description field, which is in LOG_MSG_DESC_REC.
LOG_MSG_SEVERITY	Not null	NUMBER	1	Severity level. See “IL_SEV: Severity Level Definitions.”
LOG_REP_MODE	Not null	CHAR	1	Logging level. See “IL_REP_MODE: Logging Level Definitions.”
LOG_MSG_VERSION	Not null	NUMBER	3	Message version number.
LOG_MSG_MNEMONIC	Null OK	CHAR	40	Name correlating a message to the software module that defines it.
LOG_MSG_PARSE_FNC	Null OK	CHAR	40	Path to a script or executable file that the Central Collector runs when it stores the message in the Log Central database.
LOG_MSG_TRAP_ID	Null OK	NUMBER	6	ID number for the SNMP trap.
LOG_MSG_TRAP_ENABLED	Null OK	NUMBER	1	SNMP trap flag. See “IL_TRAP_CLASS: Trap Classes.”
LOG_MSG_AUTO_ACK	Not null	NUMBER	1	Flag that indicates whether or not to automatically acknowledge the message.
LOG_MSG_DESC_REC	Not null	LONG		Description and recommendation in the following format: ■ 5 digits representing the length of the description ■ Description ■ Recommendation, which occupies the rest of the field

IL_REP_MODE: Logging Level Definitions

Table E-2 IL_REP_MODE

Field	Null?	Type	Size	Description
LOG_REP_MODE_ID	Not null	CHAR	1	Logging level. Possible values: N, D, V, S.
LOG_REP_MODE_DESC	Not null	CHAR	7	String that describes the logging level. Possible values: Normal, Debug, Verbose, Special.

IL_SEV: Severity Level Definitions

The following table describes the fields in the IL_SEV database table.

Table E-3 IL_SEV

Field	Null?	Type	Size	Description
LOG_SEV_ID	Not null	NUMBER	1	Severity level. Possible values: 1, 2, 3, 4, 5.
LOG_SEV_DESC	Not null	CHAR	13	String that describes the severity level. Possible values: Informational, Warning, Minor, Major, Critical.
LOG_SEV_TRAP_ID	Not null	NUMBER	1	ID number for the SNMP trap, based on message severity. Possible values: Any 6-digit number.
LOG_SEV_TRAP_ENABLED	Not null	NUMBER	1	SNMP trap flag, based on message severity. Possible values: 0, 1.

IL_SUBS: Subsystem Definitions

The following table describes the fields in the IL_SUBS database table.

Table E-4 IL_SUBS

Field	Null?	Type	Size	Description
LOG_SUBS_NAME	Not null	CHAR	8	Subsystem name.
LOG_SUBS_DESC	Null OK	CHAR	40	String that describes the subsystem name.

IL_SM_LOG_TABLE: Message Header Definition

The following table describes the fields in the IL_SM_LOG_TABLE database table. For more information about the message format, see “Message Header” in Appendix A, “Log Central Message and Message Definition Formats.”

Table E-5 IL_SM_LOG TABLE

Field Name	Null?	Type	Size	Description
MSG_KEY_TS	Not null	DATE		Time stamp indicating when the message was logged.
MSG_KEY_CTR	Not null	NUMBER	3	Counter for multiple messages received in the same second.
MSG_LOG_ID	Null OK	CHAR	3	Log ID.
MSG_REP_MODE	Not null	CHAR	1	Logging level. See “IL_REP_MODE: Logging Level Definitions.”

Table E-5 IL_SM_LOG TABLE

Field Name	Null?	Type	Size	Description
MSG_SEVERITY	Not null	NUMBER	1	Severity level. See “IL_SEV: Severity Level Definitions.”
MSG_SS_NAME	Not null	CHAR	8	Subsystem name. “IL_SUBS: Subsystem Definitions.”
MSG_ID	Not null	NUMBER	5	Message ID.
MSG_HOST_NAME	Not null	CHAR	64	Name of the sending host. This value can include dots, underscores, and dashes.
MSG_PID	Not null	NUMBER	5	Process ID.
MSG_UID	Not null	CHAR	8	User ID.
MSG_FCT_NAME	Null OK	CHAR	40	Name of the function.
MSG_TRAN_KEY	Null OK	CHAR	21	Entity.
MSG_VERSION	Not null	NUMBER	3	Message body version number. This is a reserved field.
MSG_TEXT	Null OK	LONG	2000	Message body.

IL_TRAP_CLASS: Trap Classes

The following table describes the fields in the IL_TRAP_CLASS database table.

Table E-6 IL_TRAP_CLASS

Field Name	Null?	Type	Size	Description
LOG_TRAP_CLASS_NAME	Not null	CHAR	16	Name of the SNMP trap class. Possible values: `message`, `severity`.
LOG_TRAP_CLASS_ENABLED	Not null	NUMBER	1	Flag that indicates whether or not to enable SNMP traps in the specified class. Possible values: `0` and `1`.

IL_USER: User Data

The following table describes the fields in the IL_USER database table.

Table E-7 IL_USER TABLE

Field Name	Null?	Type	Size	Description
USERNAME	Not null	CHAR	10	User name.
PASSWORD	Not null	CHAR	10	Password.
FIELDS	Null OK	CHAR	100	List of fields to display for this user in the Message Browser. For information about the Message Browser, see the Log Central Online Help.
ORDERS	Null OK	CHAR	100	Order in which to display the fields for this user in the Message Browser. For information about the Message Browser, see the Log Central Online Help.

F Initialization File

The default Log Central initialization file is `install_dir/etc/msg_processor.ini`, where *`install_dir`* is the directory where you installed Log Central. You can specify a different file when you run `lc_config`. For the `lc_config` command syntax, see Appendix B, “Commands.” If you use a nondefault value for the name of the initialization file, you need to set `INIFILE` in the messaging configuration file. For a description of `INIFILE`, see Appendix G, “Configuration Files.” The following table describes the properties in the initialization file.

Table F-1 Initialization File

Property	Description	Processes That Use This Value

LC.URL	URL for database
	Format: String
	Examples:
	■ MS SQL Server or Oracle Type-1 driver:
	`jdbc:odbc:mngrdb`
	■ Oracle8 Type-2 driver:
	`jdbc:oracle:oci8:@lcdbserv`
	■ Oracle Type-4 driver:
	`jdbc:oracle:thin:@amazon:1521:amazonProd`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

Table F-1 Initialization File

Property	Description	Processes That Use This Value

LC.driver	JDBC Driver Name
	Format: String
	Examples:
	■ MS SQL Server or Oracle Type-1 driver:
	`sun.jdbc.odbc.JdbcOdbcDriver`
	■ Oracle Type-2 or Type-4 driver:
	`oracle.jdbc.driver.OracleDriver`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

LC.userName	Database login name
	Format: String
	Default: `scott`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

LC.password	Database password
	Format: String
	Default: `tiger`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

Table F-1 Initialization File

Property	Description	Processes That Use This Value

LC.DBVendor	Database vendor name
	Format: String
	Possible values: `ORACLE`, `MSSQL`
	Default: `ORACLE`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

LC.parseFunction	Flag that indicates whether or not to process the parse function Format: String Possible values: `true`, `false` Default: `true`	`msg_processor`
LC.loadDB	Flag that indicates whether or not the database is available Format: String Possible values: `true`, `false` Default: `true`	`msg_processor`
LC.maxOpenTries	Maximum number of attempts to open a file Format: Number Default: `10000`	`msg_processor`
LC.openSleepTime	Sleep time, in milliseconds, if the file is not available to open Format: Number Default: `2000`	`msg_processor`
LC.readSleepTime	Sleep time, in milliseconds, between two reads if no new records Format: Number Default: `2000`	`msg_processor`

Table F-1 Initialization File

Property	Description	Processes That Use This Value
LC.readDelayTime	Delay time, in milliseconds, between two reads. This value should be `0`. Format: Number Default: `0`	`msg_processor`
LC.realTimeSleep Time	Sleep time, in milliseconds, between two reads from shared memory if no new records are available Format: Number Default: `1000`	`msg_processor`

LC.ILLog	Path for error/debug logs created by the Information Logger (IL) system
	Format: String
	Possible values: Any valid filename
	Default: `LCLog`
		`msg_processor`
		`lc_create_schema`
		`lc_drop_schema`
		`lc_user_create`
		`lc_user_delete`
		`lc_user_modify`
		`lc_user_list`
		`msgdef_export`
		`msgdef_import`
		`msgdef_delete`
		`subsystem_create`
		`subsystem_delete`

LC.Server.Parameters .MaxConnections	Maximum number of simultaneous connections. `0` means no limit. Format: Number Default: `100`	`msg_processor`

LC.Server.dbPort	Port number on which the Message Processor listens for requests from the Log Central Console
	Format: Number
	Default: `7001`
		`msg_processor,`
		`lc_launch`

G Configuration Files

The following sections describe the Log Central configuration files:

- Messaging Configuration File
- Trap Configuration File

Messaging Configuration File

The following sections describe the messaging configuration file:

- Overview of the Messaging Configuration File
- Example Messaging Configuration File
- LC_GLOBAL
- MANAGED_NODE
- DEFINE_FILTER
- DEFINE_MSG_MAPPINGS
- DEFINE_LOG_MONITOR

Overview of the Messaging Configuration File

The default Log Central messaging configuration file is *install_dir*/etc/messaging.conf, where *install_dir* is the directory where you installed Log Central. You can specify a different file when you run lc_config, which is described in Appendix B, “Commands.” If you use a nondefault location, set the BEA_LC_HOST_CONF environment variable, which is described in Appendix C, “Environment Variables.”

The messaging configuration file must include the LC_GLOBAL entry and can also include the following optional entries:

- MANAGED_NODE
- DEFINE_FILTER
- DEFINE_MSG_MAPPINGS
- DEFINE_LOG_MONITOR

Do not use tabs in the messaging configuration file. They can cause the character positions reported to be inaccurate.

Example Messaging Configuration File

```
DEFINE_FILTER "BankTrap"
if (MSGID == 11)
    {
    TRAPID = 5001
    }

DEFINE_FILTER "DropInfo"
if (MSGID == 8)
    {
    REMOTE = "NO"
    }

DEFINE_FILTER "Notify"
if (MSGBODY >= "network")
    {
    COMMAND = "/usr/mybin/notify_admin"
    }
```

```
LC_GLOBAL
    {
    CENTRAL_HOST = "MyHost"
    LOGPREFIX = "/usr/lclog"
    BACKUP_HOST = "MyBackup"
    BACKUP_LOGPREFIX = "/usr/backuplog"
    FILTER = "BankTrap"
    }

MANAGED_NODE
    {
    HOSTNAME = "MyNode"
    FILTER = "DropInfo"
    FILTER = "Notify"
    GLOBAL_FILTER = "NO"
    LOG_MONITOR="MyLogMonitor"
    }

DEFINE_MSG_MAPPINGS "MyMappings"
    {
    MAPPING = "-S |! -o %F8 -p sony -b %F12 -T %F10"
    MAPPING = "-S |! -I %F6 -u %F7 -b %F11 -x error"
    MAPPING = "-S |! -m %F3%V=%C30S| -n %F8%F10 -b %F11 -D %F2"
    }

DEFINE_LOG_MONITOR "MyLogMonitor"
    {
    LOG_FILENAME = "MyLogFile.txt"
    MSG_MAPPING = "MyMappings"
    MONITORING_INTERVAL = 2
    FIRST_MATCH = "YES"
    SELECT_PATTERN = "error"
    DISCARD_PATTERN = "ttyp0"
    }
```

LC_GLOBAL

At the minimum, the messaging configuration file must contain an LC_GLOBAL entry with CENTRAL_HOST and LOGPREFIX statements. The LC_GLOBAL entry configures parameters for the central host and all managed nodes. The following table describes the statements in an LC_GLOBAL entry.

Table G-1 LC_GLOBAL Statements

Statement	Data Type	Optional	Description
BACKUP_HOST	String	Optional	Name of the machine that the secondary Central Collector runs on. This value can include dots, underscores, and dashes.
BACKUP_LOGPREFIX	String	Optional	LOGPREFIX value for the secondary Central Collector. If this directory is on a remote file system, Log Central performance can be adversely affected.
CENTRAL_HOST	String	Not optional	Name of the central host. This value can include dots, underscores, and dashes.
ESCALATION_DIR	String	Optional	Directory for temporary log files on managed nodes. If this directory is on a remote file system, Log Central performance can be adversely affected. On UNIX, the default is /tmp. On Windows NT, the default is the directory specified by the TMP environment variable, which usually specifies C:\tmp. To determine the value of TMP on your system, run the SET TMP command.
FILTER	String	Optional	Name of a global filter. You can include multiple FILTER statements. For information about filtering, see Chapter 7, “Creating Filters.”
INIFILE	String	Optional	Log Central initialization file. The default is *install_dir*/etc/msg_processor.ini, where *install_dir* is the directory where you installed Log Central. You can specify a different file when you run lc_config. For the command syntax, see Appendix B, “Commands.” For a description of the initialization file, see Appendix F, “Initialization File.”

Table G-1 LC_GLOBAL Statements

Statement	Data Type	Optional	Description
`IPCKEY`	String	Optional	Interprocess communication (IPC) key for Log Central. If you are running multiple Log Central systems, each system needs a different value for its IPC key. `IPCKEY` must be the same as BEA_LC_IPCKEY. The default is `0xeeee0000`. For more information about BEA_LC_IPCKEY, see Appendix C, "Environment Variables."
`LOGPREFIX`	String	Not optional	Directory and file prefix for the intermediate files that the Message Receiver creates. If the directory is on a remote file system, Log Central performance can be adversely affected.
`TALK_SERVICE`	String	Optional	Communication service that the Message Senders and Message Receiver use to communicate with each other. The default is `lc_talk`. For more information about how to configure communication services, see Chapter 3, "Configuring the Central Host," and Chapter 4, "Configuring Multiple Instances of Log Central."

MANAGED_NODE

The MANAGED_NODE entry is a list of filter assignments for a managed node. A MANAGED_NODE entry is required only if you want to specify non-global filters or if you want to start Log Monitors automatically when you start Log Central. No more than one MANAGED_NODE entry may be specified for each managed node. The following table describes the statements in a MANAGED_NODE entry.

Table G-2 MANAGED_NODE Statements

Statement	Data Type	Optional	Description
FILTER	String	Optional	Name of a local filter. You can include multiple FILTER statements. For information about filtering, see Chapter 7, “Creating Filters.”
GLOBAL_FILTER	String	Optional	Indicates whether or not to turn off global filters for the managed node. Possible values: YES = Turn on global filters for this node. NO = Turn off global filters for this node. The default is NO.
HOSTNAME	String	Not optional	Central host for this managed node. This value can include dots, underscores, and dashes.
LOG_MONITOR	String	Optional	Name of a Log Monitor defined by a DEFINE_LOG_MONITOR entry. The MANAGED_NODE entry can contain multiple LOG_MONITOR statements. Include this statement only for Log Monitors that you want to start automatically when you start Log Central with the start_messaging command, which is described in Appendix B, “Commands.”

DEFINE_FILTER

For information about the DEFINE_FILTER statement, see Chapter 7, “Creating Filters.”

Note: The maximum number of filters that you can define is 50. If the messaging configuration file contains more than 50 filters, the Log Central behavior becomes unpredictable.

DEFINE_MSG_MAPPINGS

The DEFINE_MSG_MAPPINGS entry is a list of log mappings. For information about log mappings, see Chapter 5, “Creating Log Mappings.”

The following rules apply to the name of the DEFINE_MSG_MAPPINGS entry. The entry:

- Must be unique.
- Cannot be one of the predefined log mapping names. For a list of the predefined log mapping names, see Appendix H, “Predefined Log Mappings.”
- Can be up to 64 characters long.

The following rules apply to each MAPPING statement. The statement:

- Must consist of mapping options as described in Chapter 5, “Creating Log Mappings.”
- Can be up to 1024 characters long.
- Cannot contain a new line character (\n).
- Cannot be a duplicate of another MAPPING statement in the same DEFINE_MSG_MAPPINGS entry.

DEFINE_LOG_MONITOR

The DEFINE_LOG_MONITOR entry configures parameters for a Log Monitor.

The following rules apply to the name of the DEFINE_LOG_MONITOR entry:

- It is the entity name that the Log Monitor uses to register with the proc_monitor process. Each Log Monitor on one managed node must have a unique entity name. Log Central will ignore multiple DEFINE_LOG_MONITOR entries that have the same name.
- It can be up to 64 characters long.

The following table describes the statements in a DEFINE_LOG_MONITOR entry.

Table G-3 DEFINE_LOG_MONITOR Statements

Statement	Data Type	Optional	Description
LOG_FILENAME	String	Not optional	Name of the log file that the Log Monitor is monitoring. It can be up to 256 characters long.
MSG_MAPPING	String	Not optional	Name of the DEFINE_MSG_MAPPINGS entry to use.
MONITORING_ INTERVAL	Integer	Optional	Length of time to wait between forwarding each log message. The default is one second. Use 0 to forward once and then stop. Using 0 causes the Log Monitor to run as a daemon.
FIRST_MATCH	String	Optional	Flag that indicates whether or not to apply the log mappings to a log message only up to the first match. Possible values: YES = Apply the log mappings only up to the first match. NO = Apply the log mappings to the entire log message. The default is NO.
SELECT_PATTERN	String	Optional	Tells the Log Monitor to forward a log message only if it contains the specified pattern. This value is the same as the -p mapping option which is described in Chapter 5, “Creating Log Mappings.”

Table G-3 DEFINE_LOG_MONITOR Statements

Statement	Data Type	Optional	Description
`DISCARD_PATTERN`	String	Optional	Tells the Log Monitor to discard a log message if it contains the specified pattern. The message can still be forwarded if it satisfies another mapping in the messaging configuration file. This value is the same as the `-x` mapping option which is described in Chapter 5, “Creating Log Mappings.”

Trap Configuration File

The following sections describe the trap configuration file:

- Overview of the Trap Configuration File
- Example Trap Configuration File
- TRAP_HOST

Overview of the Trap Configuration File

On UNIX, the default Log Central trap configuration file is `/etc/lc_trap.conf`. On Windows NT, the default Log Central trap configuration file is `C:\etc\lc_trap.conf`. If you use a nondefault location, set the BEA_LC_TRAP_CONF environment variable, which is described in Appendix C, “Environment Variables.” For information about SNMP management, see Chapter 8, “Integrating SNMP.”

The trap configuration file consists of `TRAP_HOST` entries.

Example Trap Configuration File

```
TRAP_HOST   snmp_mgr_host1   162   public
TRAP_HOST   snmp_mgr_host2   183   public
```

TRAP_HOST

The `TRAP_HOST` entry configures a destination for SNMP traps. The trap configuration file requires a `TRAP_HOST` entry for each destination. If you do not define any trap destinations, Log Central uses the local host as the default destination.

The following table describes the values in a `TRAP_HOST` entry.

Table G-4 TRAP_HOST Values

Value	Data Type	Optional	Description
host_name	String	Not optional	Name of the destination machine for the trap notifications. This value can include dots, underscores, and dashes.
port	Integer	Not optional	The port number of the destination machine.
community	String	Not optional	Relationship between an SNMP agent and a set of SNMP managers that defines authentication, access control, and proxy characteristics. Standard values are: `PUBLIC` (read-only) `ADMIN` (read-write)

H Predefined Log Mappings

The following sections describe the predefined log mappings that Log Central provides:

- Names and Commands for Predefined Log Mappings
- BEA Tuxedo Log and BEA WebLogic Enterprise Log
- LM Predefined Mapping
- Oracle Alert Log
- Windows NT Event Log

For more information about log mappings, see Chapter 5, “Creating Log Mappings.”

Names and Commands for Predefined Log Mappings

The following table lists the types of logs for which Log Central provides predefined mappings along with the mapping names and the commands for performing the mappings.

Table H-1 Names and Commands for Predefined Log Mappings

Log	Name	Command
BEA Tuxedo Log and BEA WebLogic Enterprise Log	TUXEDO	`log_monitor -P TUXEDO -i ULOG.`*`date`*
LM Predefined Mapping	LM	`log_monitor -P LM -i` *`LM_log_file`*
Oracle Alert Log	ORACLE	`log_monitor -P ORACLE -i` *`alert_log_file`*
Windows NT Event Log	NTEVENT	`log_monitor -P NTEVENT`

BEA Tuxedo Log and BEA WebLogic Enterprise Log

The following table shows the predefined mapping of a BEA Tuxedo or BEA WebLogic Enterprise log to a Log Central message.

Table H-2 Mapping for a BEA Tuxedo or BEA WebLogic Enterprise Log

Field in Log Central Message	Value from BEA Tuxedo or BEA WebLogic Enterprise Log
Log ID	Log ID
Logging Level	Logging Level

Table H-2 Mapping for a BEA Tuxedo or BEA WebLogic Enterprise Log

Field in Log Central Message	Value from BEA Tuxedo or BEA WebLogic Enterprise Log
Date and Time	Date in log filename and time in log file
Subsystem	`LIBTUX`
Message ID	Message ID
Host	Host where the `log_monitor` process is running
Process ID	PID of `log_monitor`
User ID	User name of the `log_monitor` process
Function	Function
Entity	`0`
Body	Multiline message in the log

LM Predefined Mapping

For information about the LM predefined mapping, see the `log_monitor` command in Appendix B, "Commands."

Oracle Alert Log

The following table shows the predefined mapping of an Oracle alert log to a Log Central message.

Table H-3 Mapping for an Oracle Alert Log

Field in Log Central Message	Value from Oracle Alert Log
Log ID	Log ID
Logging Level	Logging Level
Date and Time	Date
Subsystem	`ORACLE`
Message ID	`999`
Host	Host where the `log_monitor` process is running
Process ID	PID of `log_monitor`
User ID	User name of the `log_monitor` process
Function	None
Entity	`0`
Body	Multiline message in the log

Windows NT Event Log

The following table shows the predefined mapping of a Windows NT event log to a Log Central message.

Table H-4 Mapping for a Windows NT Event Log

Field in Log Central Message	Value from Windows NT Event Log
Log ID	Log ID
Logging Level	Logging Level
Date and Time	NT Event date + Time
Subsystem	NT Event Source
Message ID	NT Event ID
Host	NT Event Computer
Process ID	PID of `log_monitor`
User ID	NT Event User
Function	None
Entity	`0`
Body	NT Event description

Index

A

acknowledgment
- in database schema E-2
- in message definition file 6-2
- in messages A-14

architecture 1-3

B

backup Central Collector. *See* secondary Central Collector.

BACKUP_HOST G-4

BACKUP_LOGPREFIX G-4

BEA Tuxedo log
- forwarding B-10
- predefined mapping H-2

BEA_LC_CONF_SERVICE
- environment variable C-2
- setting for multiple instances 4-3

BEA_LC_HOST_CONF C-2

BEA_LC_IPCKEY
- environment variable C-3
- setting for multiple instances 4-3

BEA_LC_MONITOR_WAKEUP_INTVL C-3

BEA_LC_PROC_MAX_RESTARTS C-3

BEA_LC_PROC_RESTART_INTVL C-4

BEA_LC_TALK_SERVICE C-4

BEA_LC_TRAP_CONF C-4

beaTrap D-1

beaTrapLcHost D-4

beaTrapLcLogLevel D-2

beaTrapLcMessageBody D-6

beaTrapLcMid D-4

beaTrapLcPid D-4

beaTrapLcSeverity D-6

beaTrapLcSubsys D-3

beaTrapLcTimestamp D-3

beaTrapLcTxKey D-5

beaTrapLcUid D-5

beaTrapLcVersion D-6

C

Central Collector
- overview 1-7
- starting 2-3
- stopping B-21
- *See also* secondary Central Collector.

central host
- configuration utility B-2
- configuring 3-1
- defining with lc_launch B-5
- in configuration file G-4
- *See also* host name.

CENTRAL_HOST G-4

communication services. *See* IPC key, ports, and UDP services.

configuration files
- *See* messaging configuration file *and* trap configuration file.

console. *See* Log Central Console.

customer support contact information ix

D

Data Collection Agents
- overview 1-5
- starting 2-4
- stopping B-21

database
- configuring 2-2
- in initialization file F-1
- installing 2-1
- values in configuration utility 3-2

database schema
- creating B-4
- dropping B-4
- format E-1

database users
- adding B-6
- changing B-8
- creating B-6
- creating for Log Central 2-2
- deleting B-7
- in database schema E-6
- listing B-9

date
- in database schema E-4
- in mappings 5-4
- in messages A-5
- in MIB D-3

DEFINE_FILTER 7-2

DEFINE_LOG_MONITOR G-8

DEFINE_MSG_MAPPINGS G-7

description
- in database schema E-2
- in message definition file 6-2
- in messages A-12

DISCARD_PATTERN G-9

documentation, where to find it viii

E

entity
- in filters 7-4
- in mappings 5-4
- in messages A-9

environment variables C-1

ESCALATION_DIR G-4

executable
- in database schema E-2
- in filters 7-6
- in message definition file 6-2
- in messages A-13

F

fault tolerance 1-12

FIELDS E-6

FILTER
- in LC_GLOBAL G-4
- in MANAGED_NODE G-6

filters
- creating 7-1
- defining 7-2
- description 7-1
- global 7-9
- local 7-10
- turning off 7-10

FIRST_MATCH G-8

function
- in database schema E-5
- in filters 7-4
- in mappings 5-4
- in messages A-8
- in MIB D-5

G

GLOBAL_FILTER G-6

H

host name
- configuring 3-1
- in configuration file G-6
- in database schema E-5
- in filters 7-4
- in mappings 5-4
- in messages A-6
- in MIB D-4

HOSTNAME G-6

I

IL_MSG E-2
IL_REP_MODE E-3
IL_SEV E-3
IL_SM_LOG TABLE E-4
IL_SUB E-4
IL_TRAP_CLASS E-6
IL_USER TABLE E-6
INIFILE
- for multiple instances 4-2
- in configuration file G-4

initialization file
- defining in configuration file G-4
- defining with lc_config B-2
- format F-1

intermediate files
- configuring 3-1
- directory G-5
- displaying B-15
- overview 1-7
- recovering B-12

INTERVAL 7-7
IPC key
- defining with BEA_LC_IPCKEY C-3
- in configuration file G-5

IPCKEY
- for multiple instances 4-2
- in configuration file G-5

J

JDBC driver
- in configuration utility 3-2
- in initialization file F-2
- installing 2-1

L

lc_conf. *See* UDP services.
lc_config B-2
lc_create_schema B-4
lc_drop_schema B-4
LC_GLOBAL
- for multiple instances 4-2
- for secondary Central Collector 3-4
- in configuration file G-4

lc_launch B-5
lc_talk. *See* UDP services.
lc_trap.conf G-9
lc_user_create B-6
lc_user_list B-9
lc_user_modify B-8
Log Central Console
- overview 1-13

Log Central Traps MIB D-1
log ID
- in database schema E-4
- in messages A-4

log mappings. *See* mappings.
log messages
- forwarding 7-6

log messages. *See* messages.
Log Monitor
- configuring G-8
- log_monitor B-10
- overview 1-6
- starting 2-4

LOG_FILENAME G-8
LOG_MONITOR G-6
LOG_MSG_AUTO_ACK E-2
LOG_MSG_DESC_REC E-2

LOG_MSG_ID E-2
LOG_MSG_MNEMONIC E-2
LOG_MSG_PARSE_FNC E-2
LOG_MSG_SEVERITY E-2
LOG_MSG_TRAP_ENABLED E-2
LOG_MSG_TRAP_ID E-2
LOG_MSG_VERSION E-2
LOG_REP_MODE E-2
LOG_REP_MODE_DESC E-3
LOG_REP_MODE_ID E-3
LOG_SDESC E-2
LOG_SEV_DESC E-3
LOG_SEV_ID E-3
LOG_SEV_TRAP_ENABLED E-3
LOG_SEV_TRAP_ID E-3
LOG_SUBS_DESC E-4
LOG_SUBS_NAME
  in IL_MSG E-2
  in IL_SUBS E-4
LOG_TRAP_CLASS_ENABLED E-6
LOG_TRAP_CLASS_NAME E-6
logging level
  definitions E-3
  in database schema (message header) E-4
  in database schema (message) E-2
  in filters 7-4
  in mappings 5-4
  in messages A-4
  in MIB D-2
LOGPREFIX G-5

M

managed node, configuring G-6
MANAGED_NODE G-6
mappings
  BEA Tuxedo log H-2
  configuring G-7
  date formats 5-5
  dates 5-8
  defining B-10
  description 5-2
  example (1) 5-2
  example (2) 5-12
  field lengths 5-11
  format symbols 5-7
  metacharacters 5-9
  multiple 5-3
  option values 5-6
  options 5-3
  Oracle alert log H-4
  predefined H-1
  separators 5-10
  where to define 5-3
  Windows NT event log H-5
maximum messages, defining with lc_launch B-5
message body
  in database schema E-5
  in filters 7-4
  in mappings 5-4
  in messages A-9
  in MIB D-6
message definitions
  copying B-14
  creating 6-1
  description 6-1
  example 6-3
  example of format A-2
  file, creating 6-3
  file, format 6-2
  format A-10
  loading B-15
  predefined 6-1
message header A-3
message ID
  in database schema (message header) E-5
  in database schema (message) E-2
  in filters 7-4
  in mappings 5-4

- in message definition file 6-2
- in messages A-6
- in MIB D-4

Message Processor 1-8
Message Receiver 1-8
Message Sender 1-6
messages
- construction 1-8
- dropping 7-6
- example of format A-2
- format A-1
- in database schema E-1
- saving 7-6

messaging configuration file
- compiling B-17
- defining with BEA_HOST_CONF C-2
- defining with lc_config B-2
- displaying B-17
- example G-2
- format G-1
- overview G-2

messaging.conf G-2
MIBs
- beaTrap (Log Central Traps) D-1
- description 8-1

MONITORING_INTERVAL G-8
MSG_FCT_NAME E-5
MSG_HOST_NAME E-5
MSG_ID E-5
MSG_KEY_CTR E-4
MSG_KEY_TS E-4
MSG_LOG_ID E-4
MSG_MAPPING G-8
MSG_PID E-5
msg_processor. *See* Message Processor.
msg_processor.ini F-1
msg_reader B-15
msg_receiver. *See* Message Receiver.
MSG_REP_MODE E-4
msg_sender. *See* Message Sender.
MSG_SEVERITY E-5
MSG_SS_NAME E-5
msg_test B-16
MSG_TEXT E-5
MSG_TRAN_KEY E-5
MSG_UID E-5
MSG_VERSION E-5
msgdef_delete B-13
msgdef_export B-14
msgdef_import B-15
multiple instances 4-2

O

OCCURRENCES 7-7
Oracle alert log
- predefined mapping H-4

ORDERS E-6

P

PASSWORD E-6
PATH
- for Web browser 2-4

performance data, printing B-16
ports
- configuring 3-1
- defining with lc_launch B-5
- in initialization file F-4

predefined mappings H-1
printing product documentation ix
proc_monitor. *See* Process Monitor.
process ID
- in database schema E-5
- in filters 7-4
- in mappings 5-4
- in messages A-7
- in MIB (message) D-4

Process Monitor 1-11

R

recommendation
  in database schema E-2
  in message definition file 6-2
  in messages A-12
related information ix
restarts
  BEA_LC_PROC_MAX_RESTARTS C-3
  BEA_LC_PROC_RESTART_INTVL C-4

S

secondary Central Collector
  configuring 3-3
  in configuration file G-4
  overview 1-12
  See also Central Collector.
SELECT_PATTERN G-8
severity
  definitions E-3
  in database schema (message header) E-5
  in database schema (message) E-2
  in message definition file 6-2
  in messages A-11
  in MIB D-6
shared memory, displaying B-17
show_config B-17
SNMP
  description 8-1
  integration 1-12
  management 8-2
  OIDs 8-2
SNMP traps
  advanced 1-12
  basic 1-12
  classes E-6
  configuring G-9
  in filters 7-6
  See also trap ID.
start_messaging B-19
stop_messaging B-21
subsystem
  creating B-21
  definitions E-4
  deleting B-22
  in database schema (message header) E-5
  in database schema (message) E-2
  in filters 7-4
  in mappings 5-4
  in message definition file 6-2
  in messages A-5
  in MIB (for message) D-3
summary
  in database schema E-2
  in message definition file 6-2
  in messages A-11
support
  technical x

T

TALK_SERVICE
  for multiple instances 4-2
  in configuration file G-5
temporary files
  description 1-12
  directory G-4
  recovering B-12
test messages B-16
timestamp. *See* date.
transaction ID
  in database schema E-5
  in MIB D-5
trap configuration file
  defining G-9
  defining location C-4
  example G-10
  format G-9

overview G-9

trap enabled

in database schema E-2

in message definition file 6-2

in messages A-14

See also SNMP traps.

trap ID

in database schema E-2

in message definition file 6-2

in messages A-13

See also SNMP traps.

Tuxedo log

forwarding B-10

predefined mapping H-2

U

UDP services

BEA_LC_CONF_SERVICE C-2

BEA_LC_SERVICE C-4

for central host 3-4

for multiple instances 4-2

in configuration file G-5

user ID

in database schema E-5

in filters 7-4

in mappings 5-5

in messages A-7

in MIB (for message) D-5

USERNAME E-6

W

Web browser, defining B-5

Windows NT event log, predefined mapping H-5