



THE ENTERPRISE MIDDLEWARE SOLUTION

# BEA Manager

## Log Central Administrator's Guide

BEA Manager Log Central 4.0  
Document Edition 4.0  
November 1998

# Copyright

Copyright © 1998 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, ObjectBroker, TOP END, and TUXEDO are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, Jolt and M3 are trademarks of BEA Systems, Inc.

All other company names may be trademarks of the respective companies with which they are associated.

### BEA Manager Log Central Administrator's Guide

Document Edition	Part Number	Date	Software Version
4.0	815-001003-001	November 1998	BEA Manager Log Central 4.0

---

# Contents

## Preface

Purpose of This Document .....	xi
Who Should Read This Document .....	xi
How This Document Is Organized .....	xi
How to Use This Document .....	xiii
Opening the Document in a Web Browser .....	xiii
Printing from a Web Browser .....	xv
Documentation Conventions .....	xv
Related Documentation .....	xvii
Log Central Documentation .....	xvii
Contact Information .....	xvii
Documentation Support .....	xvii
Customer Support .....	xviii

## 1. Overview

Log Messages in System Management .....	1-1
Agent/Manager Architecture .....	1-2
Log Central and Enterprise Management Systems .....	1-3
Data Collection Agents .....	1-5
Log Central Components on the Managed Node .....	1-5
Log Monitor .....	1-7
Message Sender .....	1-8
Process Monitor .....	1-9
Central Collector .....	1-9
Message Receiver .....	1-12
Message Processor .....	1-12

---

## 2. Getting Started

Setting Up Log Central .....	2-1
Optional Setup Tasks .....	2-3
Configuring Log Central .....	2-4
Required Information for Host Configuration Utility .....	2-4
Running the Host Configuration Utility .....	2-6
Configuring Multiple Instances of Log Central .....	2-7

## 3. Configuring the Database for Use with Log Central

Managing Your Database .....	3-1
Creating a Database Schema .....	3-2
Dropping a Database Schema .....	3-2
Categorizing Messages by Resource .....	3-3
Creating Subsystem Entries in the Database .....	3-3
Deleting a Subsystem Entry .....	3-4
Managing Log Central User Entries .....	3-5
Add a User .....	3-5
Delete a User .....	3-6
Modify a Password .....	3-6
List Users .....	3-7

## 4. Integrating Logs into Log Central

Log Monitor Configuration File Options .....	4-2
Specifying Option Values .....	4-6
Example of Using Log Monitor with a Configuration File .....	4-6
Multiple Separators with the -S Option .....	4-7
Using the -p and -x Options .....	4-8
Field Lengths .....	4-10
Specifying the Date Format (%f) .....	4-11
Filtering a System Log .....	4-12
Specifying Date Format .....	4-15
Converting Input Dates .....	4-15

---

## 5. Creating and Loading Message Definitions

Message Definitions .....	5-1
Getting Message Definitions into the Log Central Database .....	5-2
Create a Message Definition File .....	5-2
Description of Message Definition File .....	5-3
Fields of a Message Definition File .....	5-4
Example .....	5-5
Load the Message Definition File .....	5-6
Using Other Message Definition Commands .....	5-7
Exporting Message Definitions .....	5-7
Deleting Message Definitions .....	5-8
How to Delete a List of Message Definitions .....	5-8
How to Delete Message Definitions Associated with a Subsystem .....	5-8

## 6. Host and Filter Configuration

Default Configuration .....	6-2
Where to Put Intermediate Files .....	6-2
Specifying a Backup Central Collector .....	6-3
Host Name Usage .....	6-3
Specifying Nondefault Global Parameters .....	6-4
Assigning Filters to Agents .....	6-6
Using a MANAGED_NODE Entry .....	6-7
Turning Off Global Filters on a Particular Node .....	6-7
Assigning a Filter to a Particular Node .....	6-8
Defining Agent Filters .....	6-9
Defining Conditions .....	6-11
Defining Actions .....	6-12
Dropping a Message .....	6-12
Executing a Command .....	6-13
Sending an SNMP Trap Notification .....	6-13
Example .....	6-14
Filter Recommendations .....	6-16

---

## 7. Integrating Log Central with an SNMP Manager

Facilities for Managing Log Central .....	7-2
Management Information Base Support for Log Central.....	7-2
Process Monitor Agent .....	7-3
Central Collector SNMP Trap Generation .....	7-3
Data Collection Agent SNMP Trap Generation .....	7-3
Setting Up SNMP Management of Log Central .....	7-4
Integrating Events Generated by Agent Integrator Polling .....	7-6

## 8. Starting and Stopping Log Central

Starting the Log Central Data Collection System .....	8-2
Starting Log Central on a Central Host .....	8-3
Starting Log Central on the Managed Nodes .....	8-5
Starting Log Monitor .....	8-6
Starting Log Monitor with Predefined Mappings .....	8-6
Starting Log Monitor with Mappings in a Configuration File .....	8-8
Starting Log Monitor with Mapping Specified on the Command Line .....	8-9
Stopping the Log Central Data Collection System.....	8-12
Displaying Log Central System Information.....	8-13
Example.....	8-14

## 9. Using the Log Central Console

Invoking the Launch Panel.....	9-1
Using the Message Browser .....	9-3
Message Browser Main Window .....	9-4
How to Monitor Incoming Messages .....	9-7
How to Perform Historical Queries of the Message Database .....	9-7
How to Delete Messages .....	9-8
How to Acknowledge Messages .....	9-9
How to Remove Acknowledgment from Messages .....	9-10
How to Filter Messages .....	9-10
How to Change the Message Layout in the Main Window.....	9-16
How to Change Message Colors .....	9-17
How to View Message Details .....	9-19

---

How to Generate Reports .....	9-19
How to View Summary Reports .....	9-19
How to View Detail Reports .....	9-22
Using the Message Definition Editor .....	9-24
Message Definition Main Window .....	9-25
How to Display Message Definitions.....	9-27
How to Add a New Message Definition .....	9-28
How to Modify a Message Definition.....	9-30
How to Delete Message Definitions.....	9-30
How to Define Filtering Criteria for Retrieving Message Definitions.....	9-32
How to Change the Message Definition Layout .....	9-33
How to Change Message Definition Colors.....	9-34
How to Modify the Subsystem Description .....	9-36
How to Generate Reports .....	9-37
How to View Summary Reports .....	9-37
How to View Detail Reports .....	9-40
Using the Basic Trap Configuration Tool .....	9-42
Configuring Trap Generation by Severity.....	9-43
Configuring Trap Generation by Message Definition.....	9-45
Generating Traps Based on both Severity and Message Definition.....	9-46
Using the Storage Maintenance Tool .....	9-47
Storage Maintenance Main Window .....	9-48
How to Prepare the Records Processing Script.....	9-50
Example .....	9-51
How to Schedule Deletion of Database Records .....	9-52
How to Manually Process Database Records.....	9-53
How to Schedule Processing of Database Records .....	9-54
How to Schedule Both Processing and Deletion of Database Records .....	9-56
How to Schedule Deletion of Intermediate Files .....	9-57

## **A. Message Format**

Message Format.....	A-2
Message Attributes at the Agent.....	A-3
Log ID .....	A-3
Logging Level .....	A-3

---

Date and Time .....	A-3
Subsystem Name .....	A-4
Message ID.....	A-4
Host Name .....	A-4
Process ID.....	A-5
User ID .....	A-5
Function Name .....	A-5
Transaction ID.....	A-5
Body .....	A-5
Attributes in the Message Definition.....	A-6
Severity.....	A-6
Summary.....	A-7
Description .....	A-7
Recommendation.....	A-7
Trap Generation.....	A-7
Trap ID .....	A-8
Automatic Acknowledgment Flag.....	A-8
Execute on DB Upload.....	A-8

## **B. Environment Variables**

BEA_LC_IPCKEY.....	B-1
Using a New IPCKEY Value .....	B-2
BEA_LC_CONF_SERVICE.....	B-3
Using a Different UDP Communication Service .....	B-3
BEA_SM_BEAMGR_CONF.....	B-4
Setting a New Location for the BEA Manager Configuration File.....	B-4

## **C. Testing and Debugging Commands**

Generating Test Messages .....	C-1
Examples .....	C-2
Reading the Current Intermediate Log File.....	C-3
Examples .....	C-4



---

## D. MIB Reference

Log Central Traps MIB .....	D-2
beaTrapLcLogLevel .....	D-3
beaTrapLcTimestamp .....	D-3
beaTrapLcSubsys .....	D-3
beaTrapLcMid .....	D-3
beaTrapLcHost .....	D-4
beaTrapLcPid .....	D-4
beaTrapLcUid .....	D-4
beaTrapLcFunction .....	D-4
beaTrapLcTxKey .....	D-5
beaTrapLcVersion .....	D-5
beaTrapLcSeverity .....	D-5
beaTrapLcMessageBody .....	D-5
Process Monitor MIB .....	D-6
beaPmProcsEnvVar .....	D-6
beaPmMonitorPid .....	D-6
beaPmMonitorTimer .....	D-7
beaPmMonitorLastWakeup .....	D-7
beaPmMaxProcRestarts .....	D-7
beaPmMaxProcRestartsIntvl .....	D-7
beaPmProcTable .....	D-8
beaPmTblEntityName .....	D-9
beaPmTblProcId .....	D-9
beaPmTblStatus .....	D-9
beaPmTblFirstRegTime .....	D-9
beaPmTblLastRegTime .....	D-10
beaPmTblRestartEnabled .....	D-10
beaPmTblRestartCmd .....	D-10
beaPmTblNumRestarts .....	D-10
beaPmTblRefRestartTime .....	D-10
beaPmTblNumRestartsFromRef .....	D-11
beaPmTblFirstRestartTime .....	D-11
beaPmTblLastRestartTime .....	D-11
beaPmTblUid .....	D-11

beaPmTblGid .....	D-11
beaPmTblEuid .....	D-12
beaPmTblEgid .....	D-12
beaPmTblSwName .....	D-12
beaPmTblSwVersion .....	D-12
beaPmTblSwDate .....	D-13
beaPmTblSwTime .....	D-13

## E. Database Schema

Schema Tables .....	E-2
Log Message Definitions .....	E-3
Logging Levels .....	E-4
Message Severities .....	E-4
Subsystem Definitions .....	E-5
Trap Classes .....	E-5
Logged Message Data .....	E-6
User Data .....	E-7

## F. Initialization File

Initialization File .....	F-1
---------------------------	-----

## G. Predefined Log Mapping

BEA TUXEDO Message Mapping .....	G-2
Log Central Message Fields .....	G-2
Windows NT Event Log .....	G-3
Oracle Alert Log .....	G-4

## Glossary

## Index

---

# Preface

## Purpose of This Document

This document describes the BEA Manager Log Central and gives instructions for using the graphical user interface for managing Log Central applications.

## Who Should Read This Document

This document is intended for administrators who manage Log Central applications, and administrators of applications that are being managed by Log Central.

## How This Document Is Organized

The *Log Central Administrator's Guide* is organized as follows:

- ◆ Chapter 1, “Overview,” provides an overview of Log Central.
- ◆ Chapter 2, “Getting Started,” describes the steps that an administrator needs to perform to set up a Log Central system.
- ◆ Chapter 3, “Configuring the Database for Use with Log Central,” describes utilities that configure the database.
- ◆ Chapter 4, “Integrating Logs into Log Central,” describes how to create mappings to map third-party log messages into Log Central log message format.

- 
- ◆ Chapter 5, “Creating and Loading Message Definitions,” describes how to create and load message definitions in a batch process.
  - ◆ Chapter 6, “Host and Filter Configuration,” describes how to configure a backup collector, and define filters.
  - ◆ Chapter 7, “Integrating Log Central with an SNMP Manager,” describes how to integrate Log Central with an SNMP Manager.
  - ◆ Chapter 8, “Starting and Stopping Log Central,” describes starting a Central Collector on the central host and starting agents on managed nodes.
  - ◆ Chapter 9, “Using the Log Central Console,” describes management tasks implemented through the tools of the Log Central graphical user interface.
  - ◆ Appendix A, “Message Format,” describes Log Central message attributes at the agent and in the message definition.
  - ◆ Appendix B, “Environment Variables,” describes certain environment variables in Log Central, and why you may need to change them.
  - ◆ Appendix C, “Testing and Debugging Commands,” describes how to generate test messages, read the current log file, and display the BEA Manager version number.
  - ◆ Appendix D, “MIB Reference,” describes the Log Central Traps MIB and the Process Monitor MIB.
  - ◆ Appendix E, “Database Schema,” details the contents of the database table definitions used by Log Central.
  - ◆ Appendix F, “Initialization File,” details the contents of the Log Central initialization file.
  - ◆ Appendix G, “Predefined Log Mapping,” details the predefined mappings from the messages of BEA TUXEDO applications, NT, and Oracle to those stored in the Log Central database.

---

# How to Use This Document

This document, *Log Central Administrator's Guide*, is designed primarily as an online, hypertext document. If you are reading this as a paper publication, note that to get full use from this document you should install and access it as an online document via a Web browser.

The following sections explain how to view this document online, and how to print a copy of this document.

## Opening the Document in a Web Browser

To access the online version of this document, open the following HTML file in a Web browser:

```
install_dir/docs/mgr/20/logcent/index.htm
```

`install_dir` is the directory in which you installed Log Central.

**Note:** The online documentation requires a Web browser that supports HTML version 3.0. Netscape Navigator version 2.02 or Microsoft Internet Explorer version 3.0 or later are recommended.

Figure 1 shows the online document with the clickable navigation bar and table of contents.

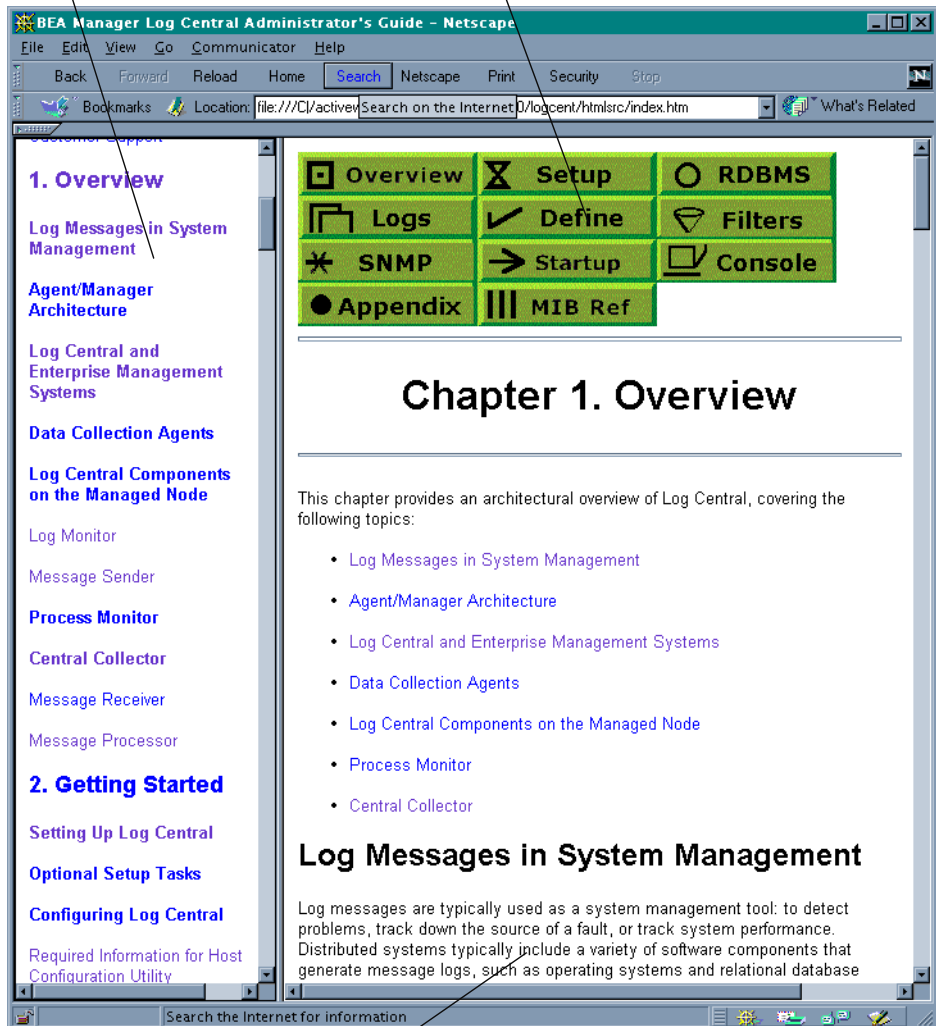
**Figure 1 Online Document Displayed in a Netscape Web Browser**

**Table of Contents**

Click a topic to view it.

**Navigation Bar**

Click a button to view a main topic.



---

# Printing from a Web Browser

You can print a copy of this document, one file at a time, from the Web browser. Before you print, make sure that the chapter or appendix you want is displayed and *selected* in your browser. (To select a chapter or appendix, click anywhere inside the chapter or appendix you want to print. If your browser offers a Print Preview feature, you can use the feature to verify which chapter or appendix you are about to print.)

The BEA Manager Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

# Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys sequentially.
<i>italics</i>	Indicates emphasis or book titles.
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. <i>Examples:</i> <pre>#include &lt;iostream.h&gt; void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>

Convention	Item
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> void <b>commit</b> ( )
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> 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. <i>Example:</i> buildobjclient [-v] [-o name ] [-f <i>file-list</i> ]... [-l <i>file-list</i> ]...
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> <li>◆ That an argument can be repeated several times in a command line</li> <li>◆ That the statement omits additional optional arguments</li> <li>◆ That you can enter additional parameters, values, or other information</li> </ul> The ellipsis itself should never be typed. <i>Example:</i> buildobjclient [-v] [-o name ] [-f <i>file-list</i> ]... [-l <i>file-list</i> ]...
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.



---

# Related Documentation

The following sections list the documentation provided with Log Central, and other related publications.

## Log Central Documentation

The Log Central information set consists of the following documents:

- ◆ *Log Central Administrator's Guide*
- ◆ Log Central online help
- ◆ *BEA Manager Installation Guide*
- ◆ *BEA Manager Release Notes*

**Note:** The BEA Manager Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

## Contact Information

The following sections provide information about how to obtain support for the documentation and software.

## Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com**. (For information about how to contact Customer Support, refer to the following section.)

---

# Customer Support

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.beasys.com](http://www.beasys.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

# 1 Overview

This chapter provides an architectural overview of Log Central, covering the following topics:

- ◆ Log Messages in System Management
- ◆ Agent/Manager Architecture
- ◆ Log Central and Enterprise Management Systems
- ◆ Data Collection Agents
- ◆ Log Central Components on the Managed Node
- ◆ Process Monitor
- ◆ Central Collector

## 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 typically include a variety of software components that generate message logs, such as operating systems and relational database management systems (RDBMS). In the absence of any standard, software makers use different practices for message logging.

Log Central allows you to extract the information from these diverse logs and map the information into a common format. The information is maintained in a single relational database, providing a single point of access and a unified view of all information contained in log messages. This database approach improves the manageability of distributed systems.

A single failure, such as a file system filled to capacity, can generate a number of different log messages as the problem ripples through the affected software components. A unified view of the various messages means that the source of a problem can be more rapidly diagnosed.

All messages are stored in an RDBMS, and users can view the logs, generate reports, and do online monitoring through a set of graphical user interface tools—called the Log Central *Console*—and through several commands offered at the operating system level.

Also, each message is associated with a message definition, which includes information such as severity (degree of impact on the distributed system), probable cause of the message generated, and actions that need to be taken when it is logged. This information can be viewed and updated online by the administrator, who can use it to form a knowledge base for resolving problems.

Log information can be monitored “in real time” as it arrives at the Central Collector, using the Log Central Message Browser (part of the Log Central Console). The Central Collector stores management information in a relational database system which can be queried for analysis of problems or to track trends.

How to use the Log Central Console is discussed in Chapter 9, “Using the Log Central Console.”

## Agent/Manager Architecture

Log Central is based on an agent/manager architecture, as shown in Figure 1-1. Local data collection agents run on machines where you have resources to be managed—these machines are called *managed nodes*. Local data collection agents forward log messages to the Central Collector. The Log Central Console, the Central Collector, and the Log Central relational database together play the “manager” role in the Log Central system.

The log agents monitor log messages generated by the resources that you wish to manage, such as messages logged to the UNIX syslog or NT event log, BEA TUXEDO userlogs, or relational database system logs. Agents map the information from these log messages into Log Central’s uniform internal format for forwarding to the Central Collector. The data collection agents can be distributed around the network as needed.

To implement fault tolerance, users can configure a secondary Central Collector. If the primary Central Collector becomes unavailable, management information is automatically sent to the secondary Central Collector. Control automatically switches back to the primary Central Collector when it becomes available. Once the primary Central Collector becomes available again, the information sent to the secondary Central Collector is available to the primary Collector if the primary and secondary Central Collectors have been configured to use the same RDBMS.

How to configure a backup Central Collector is described in Chapter 6, “Host and Filter Configuration.”

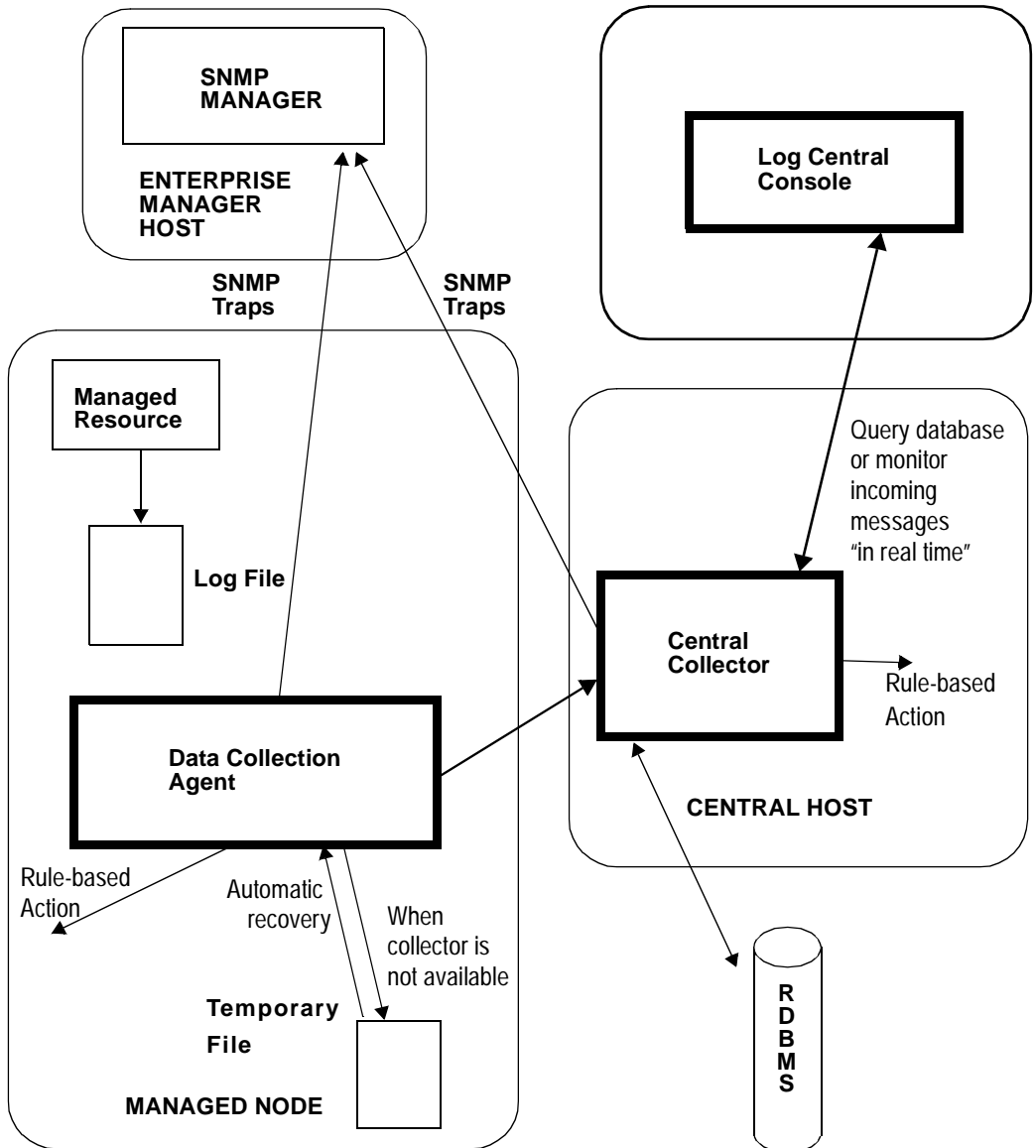
When no Central Collector is available to the data collection agent, the agent automatically stores the information in a temporary local backup file. Information in this file is automatically recovered and passed to the Central Collector when the Central Collector becomes available.

## Log Central and Enterprise Management Systems

Log Central allows you to integrate information from logs into an enterprise management system using Simple Network Management Protocol (SNMP). Both the data collection agents and the Central Collector have the ability to generate SNMP trap notifications. Two levels of trap configuration are available:

- ◆ A basic mapping of log messages to SNMP traps can be configured at the Central Collector using the Log Central Console Basic Trap Configuration window. The Central Collector generates SNMP traps when messages arrive that match the selected criteria. Using the Log Central Console Basic Trap Configuration is described in Chapter 9, “Using the Log Central Console.”
- ◆ By editing the Log Central configuration file (`messaging.conf`), more complex criteria can be specified to select messages that trigger SNMP trap notifications. These criteria allow you to generate SNMP trap notifications from the distributed data collection agents. Defining agent filters to generate SNMP traps is described in Chapter 6, “Host and Filter Configuration.”

Figure 1-1 Log Central in an Enterprise Network



Steps for integrating Log Central with your enterprise management system are described in Chapter 7, “Integrating Log Central with an SNMP Manager.”

## Data Collection Agents

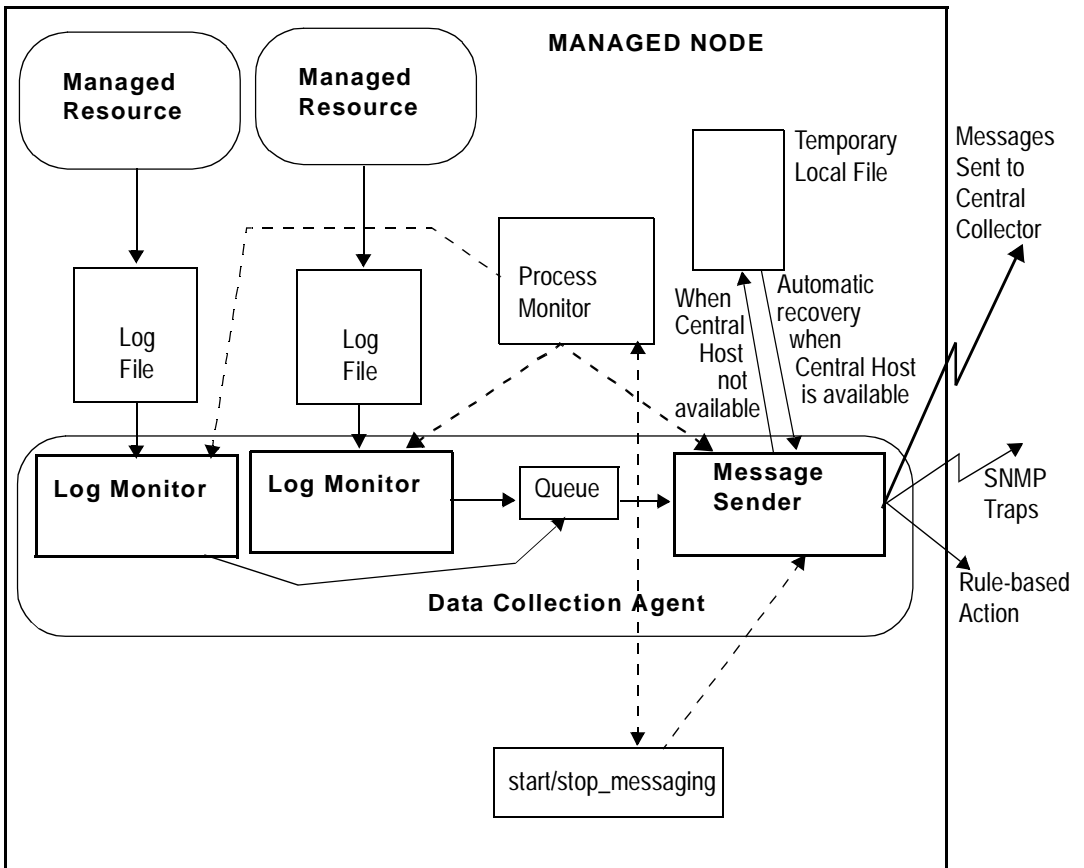
The data collection agent is made up of a Message Sender and one or more Log Monitors. Log Monitors monitor log messages generated by the resources that you wish to manage, such as messages logged to the UNIX `syslog` or Windows NT event log, BEA TUXEDO userlogs, or relational database system (RDBMS) logs. A distinct Log Monitor process is used on each managed node to monitor a particular log and map the information from incoming log messages to Log Central internal log format. The log messages are then forwarded to the Message Sender.

## Log Central Components on the Managed Node

The flow of information from the managed resource through the agent to the Central Collector is shown in Figure 1-2.

**Note:** Although two Log Monitors are shown in this diagram, the data collection agent can in fact have one or any number of Log Monitor processes. You must have a separate Log Monitor process for each log that you wish to monitor.

**Figure 1-2 Log Central Flow of Information**





## Log Monitor

The Log Monitor reads the logs generated by the managed resource, such as a computer system, a BEA TUXEDO application, or a relational database system. Log Monitor maps the attributes in the managed resource log messages to attributes in Log Central messages. Messages are then placed in the Message Sender queue for forwarding to the Central Collector. You need a dedicated Log Monitor process for each managed resource.

Mapping of information into the Log Central internal format is provided out-of-the-box for the following logs:

- ◆ BEA TUXEDO user logs
- ◆ Oracle alert logs
- ◆ Windows NT event log

Message definitions are also provided out-of-the-box for these log resources.

This is an extensible system in that other log resources can be incorporated into Log Central as well. To manage other log resources you need to provide two things:

- ◆ A *mapping* to enable Log Monitor to translate the log messages into Log Central format.

You can define different mappings for the same log file—up to 20 different Log Central messages could be generated from a single message logged by the managed resource. How to devise mappings for log files is discussed in Chapter 4, “Integrating Logs into Log Central.”

- ◆ *Message definitions* corresponding to the messages that will be forwarded by data collection agents to the Central Collector. A message definition template and command-line utilities are provided for loading message definitions when setting up the Log Central system. This is described in Chapter 5, “Creating and Loading Message Definitions.”

For a list of all the procedures for setting up Log Central, refer to Chapter 2, “Getting Started.”

## Message Sender

The Message Sender reads incoming messages from its queue and forwards them to its primary Central Collector. If the primary Central Collector is not available, the Message Sender sends the message to a secondary Central Collector if a secondary Central Collector has been defined. There should be one message sender for every managed node.

Agent filters can be defined for the Message Sender to do the following:

- ◆ Discard (not forward) specified messages
- ◆ Send a Simple Network Management Protocol (SNMP) trap notification when a specified message occurs
- ◆ Execute a script or program when a specified message occurs
- ◆ Store a specified message in a local file

Configuring filters is described in Chapter 6, “Host and Filter Configuration.”

If none of the Central Collectors configured for this agent are accessible—due to a network outage, for example—the Message Sender writes messages retrieved from its queue to a temporary local file. When the Central Collector becomes available, the Message Sender automatically recovers all messages from the temporary file and forwards them to the Central Collector. When automatic recovery is occurring, new incoming messages have the highest priority and recovered messages are forwarded when the Message Sender is not preoccupied with new messages.

Temporary files are automatically deleted once the messages have been forwarded to the Central Collector.

If the Message Sender is unable to save messages in a temporary file, the messages are discarded and an SNMP trap is generated.

For information on starting and stopping Log Central processes on the managed node, refer to Chapter 8, “Starting and Stopping Log Central.”

# Process Monitor

The Process Monitor (`proc_monitor`) is a daemon that runs on all managed nodes and the central host. The Process Monitor is started whenever the `start_messaging` command is invoked on a particular machine. The following processes “register” with the Process Monitor at startup time:

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

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

# Central Collector

The Central Collector performs the following functions:

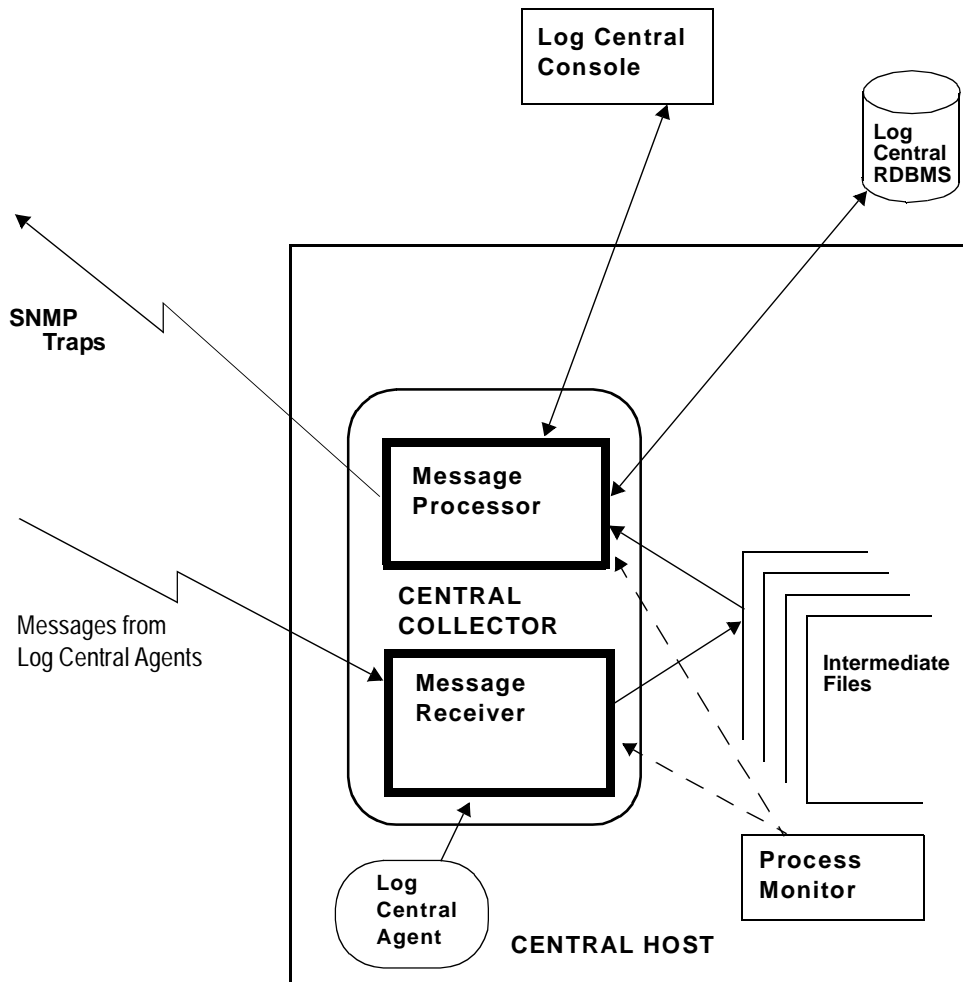
- ◆ Log information collected by the distributed data collection agents is forwarded to the Central Collector. The Central Collector is your central point of access to the stream of information flowing from distributed log agents. When you monitor incoming messages using the Log Central Message Browser, the Central Collector provides the messages to the Message Browser as they arrive from the log agents.
- ◆ The Central Collector is responsible for inserting incoming messages into the Log Central database. The Log Central database may be located either on the central host or on another machine in your network. The Central Collector is your point of access to the historical log data maintained in the database. When you make queries for historical log data through the Log Central Message Browser, the Central Collector processes your requests, retrieves the requested data from the database, and sends the data to the Message Browser.

- ◆ The Log Central Console Basic Trap Configuration window allows you to do basic configuration of SNMP trap notifications generated by the Central Collector. More advanced conditions can be defined for generating SNMP traps using agent filters. These are defined by editing the Log Central configuration file (`messaging.conf`) on the central host.
- ◆ The Central Collector is also your point of access to the message definitions maintained in the Log Central database. Requests for viewing or modifying message definitions from the Message Definition Editor are processed by the Central Collector.

For fault tolerance, you can configure two Central Collectors with one serving as the backup or secondary collector in case the primary Central Collector goes down or is unavailable.

The Central Collector is made up of two processes, the Message Receiver and the Message Processor. The flow of information at the Central Collector is shown in Figure 1-3.

Figure 1-3 Log Central Components on the Central Host



## Message Receiver

Messages from the distributed data collection agents arrive at the Message Receiver. The Message Receiver stores the incoming messages in an intermediate file. A new intermediate file is created every hour. You can control how frequently the intermediate files are deleted using the Log Central Console Storage Maintenance tool.

For information on using the Storage Maintenance tool, refer to Chapter 9, “Using the Log Central Console.”

The Message Receiver generates an enterprise-specific SNMP trap if it cannot log messages to an intermediate file. The number of failures that triggers a trap is defined by the `BEA_LC_TRAP_EVERY_FAILURES` environment variable. If this environment variable is not set, a trap is generated every 100 failures. The SNMP trap has an enterprise-specific trap number of 90101.

## Message Processor

The Message Processor performs the following functions:

- ◆ Reads messages from intermediate files and inserts these messages into the Log Central relational database.
- ◆ Handles user-initiated requests from Log Central Consoles, such as database queries or forwarding of incoming messages for online monitoring.
- ◆ Generates SNMP trap notifications if the user has enabled trap generation at the Central Collector. This can be configured using the Log Central Console Basic Trap Configuration window. Traps can be generated either on the basis of severity or the message definition. How to do Basic Trap Configuration is described in Chapter 9, “Using the Log Central Console.”

# 2 Getting Started

This chapter describes the required administrative steps to set up a Log Central system. The procedure assumes that you have already installed Log Central on the managed nodes and the central host, as described in the *BEA Manager Installation Guide*.

The following topics are included:

- ◆ Setting Up Log Central
- ◆ Optional Setup Tasks
- ◆ Configuring Log Central
- ◆ Configuring Multiple Instances of Log Central

## Setting Up Log Central

To set up Log Central, perform the following steps:

1. Ensure that the Log Central relational database is installed and accessible to the Log Central Central Collector.

For a list of supported databases, refer to the *BEA Manager Release Notes*. For database installation details, consult the database vendor's documentation. The installed database must have a Java Database Connectivity (JDBC) driver available. For details on how to install this, consult your database JDBC documentation.

2. Create a Log Central database user (for example, `lc_user`).

This user should have privileges to create and delete tables and modify their contents. Allocate a minimum of 10 Mb of disk space for the database user. Log Central uses this database user to access the database and create and update tables that store Log Central messages.

3. Install the Java Runtime Environment (JRE) on the Central Host machine.

The JRE should be available from your operating system vendor. Modify your environment so that the path includes the directory that has the JRE program.

4. On the central host, configure the Log Central system using the `lc_config` program, as described in “Configuring Log Central.”

**Note:** You need to create service entries for the UDP services—if they do not exist—`lc_talk` (default port 7012) and `lc_conf` (default port 7011). These services need to be available on the central host, and managed nodes, as well. For more information, consult your network administrator.

Service entry examples:

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

5. Create the Log Central database schema.

To do this, run the `lc_create_schema` program. For more information, refer to Chapter 3, “Configuring the Database for Use with Log Central.”

6. Create mappings for logs that you want to monitor.

Each Log Monitor process on a managed node monitors a log and maps incoming log messages to Log Central format. Mappings are provided out-of-the-box for BEA TUXEDO logs, the Oracle alert log, and the NT event log. You need to create mappings for other logs that you wish to monitor.

How to do this is described in Chapter 4, “Integrating Logs into Log Central.”

7. Create and load message definitions.

In addition to the log message attributes that are contained in the messages sent by Log Central data collection agents, further attributes for each message are contained in the message definition, which is stored in the Log Central database. The definition includes such attributes as severity and recommendation (the action recommended in response to a message). You can modify or add definitions one at a time using the Message Definition Editor—a tool in the Log Central Console. However, when initially setting up Log Central, you may want to load a number of message definitions at once. It is easier to do this as a batch process using command-line utilities provided for this purpose.

How to do this is described in Chapter 5, “Creating and Loading Message Definitions.”

**Note:** You may wish to create subsystem entries. How to do this is described in Chapter 3, “Configuring the Database for Use with Log Central.”



8. Integrate Log Central with your network or system manager.

The Log Central system itself is manageable from a management system that supports Simple Network Management Protocol (SNMP). Log Central can also be configured to send SNMP trap notifications to a system manager in response to specified log messages.

How to do this is described in Chapter 7, “Integrating Log Central with an SNMP Manager.”

9. Start Log Central components on the central host.

How to do this is described in Chapter 8, “Starting and Stopping Log Central.”

10. Start Log Central components, including the necessary Log Monitor processes, on the managed nodes.

How to do this is described in Chapter 8, “Starting and Stopping Log Central.”

11. To use Log Central to monitor log resources, invoke the Log Central Console on the Central Host using the `lc_launch` command.

How to do this is described in Chapter 9, “Using the Log Central Console.”

## Optional Setup Tasks

Optional setup tasks include:

### ◆ Setting environment variables

You might want 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. You can do so before starting Log Central. If you want to set environment variables later, you must first stop Log Central, as described in Chapter 8, “Starting and Stopping Log Central.”

Environment variables are described in detail in Appendix B, “Environment Variables.”

### ◆ Specifying agent filters

If you want to specify filters for different managed nodes, modify the configuration file appropriately. This is described in Chapter 6, “Host and Filter Configuration.”

# Configuring Log Central

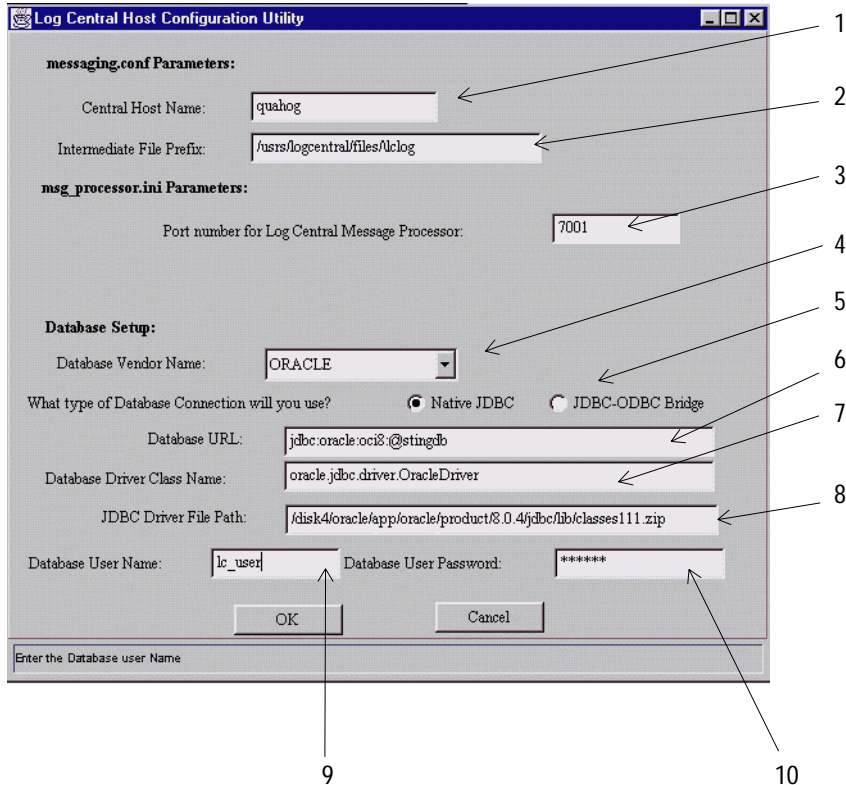
After you have ensured that all of the software components previously specified have been properly installed, and you have installed Log Central on the central host, you can proceed with configuring Log Central. To do so, you run the `lc_config` program. However, before running this program, you should be prepared to provide the information listed under “Required Information for Host Configuration Utility.”

## Required Information for Host Configuration Utility

Running the `lc_config` program brings up the form shown in Figure 2-1. You might want to fill out a worksheet first and have it handy.

The numbered list that follows the figure correspond to the numbers in Figure 2-1.

**Figure 2-1 Host Configuration Utility**



You need to ascertain the following:

1. The central host machine name  
This name should match the value returned by the `hostname` command as run on the central host.
2. The string to prefix to the names of the intermediate log files
3. The port number on which the Log Central message processor listens (the default is 7001)
4. The database name (either Oracle or MSSQL)

5. The type of database connection

For more information, refer to your database documentation.

6. The URL to access the database

For more information, refer to your database documentation.

7. The database driver class name

For more information, refer to your database documentation.

8. The complete file name to the JDBC driver (for example,  
`d:\orant\jdbc\lib\classes111.zip`)

The configuration program copies the file specified here to  
`install_dir/bin/JDBCDrvForLC.zip`

9. The Log Central database user name

This is the user you created in the second step under “Setting Up Log Central.”  
The configuration program then uses this user to create and populate the tables  
of Log Central.

10. The user password

## Running the Host Configuration Utility

After you have determined the information in the preceding section, you run the host configuration utility, as shown in the following steps.

11. Run the `lc_config` program on the central host.

The syntax for `lc_config` follows:

```
lc_config [-infile initialization_file]  
[-conffile configuration_file] [-fn fontname] [-fs fontsize]
```

The default for *initialization\_file* is  
`install_dir/etc/msg_processor.ini`

The default for *configuration\_file* is `install_dir/etc/messaging.conf`  
(where *install\_dir* is the directory in which you installed Log Central).

12. Fill out the fields of the script to correspond to those you determined earlier.

You can do more extensive configuration of Log Central than that which you have just accomplished using `lc_config`. This is described in Chapter 6, “Host and Filter Configuration.”

# Configuring Multiple Instances of Log Central

An instance of a Log Central system consists of software integrated across a central host and several managed nodes. All messages from Log Central are collected and centralized on the central host. If one set of messages generated by a set of applications is independent of another set of messages generated by another set of applications, yet they share at least one physical machine, and you need to administer both sets independently, then you must configure multiple instances of Log Central. For example, if you have more than one BEA TUXEDO domain, and you need to administer them differently, then you would create a separate instance of Log Central for each BEA TUXEDO domain.

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

1. Define two 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 use unused ports.

For example, they might have entries that look like the following:

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

2. Make a copy of the Log Central configuration file (`install_dir/etc/messaging.conf`), as, for example, `install_dir/etc/messaging.conf.dom2`.
3. Make a copy of the Log Central initialization file (`install_dir/etc/message_processor.ini`), as, for example, `install_dir/etc/message_processor.ini.dom2`.
4. Edit the copied `messaging.conf` file of step 2.

Change the `TALK_SERVICE` parameter to the newly defined service. For example, the parameter might look like the following:

```
TALK_SERVICE = "lc_talk_dom2"
```

5. In the same file, set the `IPCKEY` entry to a value different from the value of any other instance. For example:

```
IPCKEY = 0xee220000
```

`IPCKEY` can take any numeric value. Use `0x` to indicate a hexadecimal value.

6. In the same file, set the `INIFILE` entry to the copied `message_processor.ini` file in step 3. For example:

```
INIFILE = install_dir/etc/message_processor.ini.dom2
```

7. Run the `lc_config` command to configure the newly copied file. For example:

```
lc_config -conf file install_dir/etc/messaging.conf.dom2  
-ini file install_dir/etc/message_processor.ini.dom2
```

8. Set the environment variable `BEA_LC_IPCKEY` to the value set in step 5, as, for example, `0xee220000`.
9. Set the environment variable `BEA_LC_CONF_SERVICE` to one of the new service values, for example, if you are configuring a second instance, `lc_conf_dom2`.

For more information about the two preceding variables, refer to Appendix B, “Environment Variables.”

10. When you start the Central Collector, pass the copied `messaging.conf` file as a parameter. For example:

```
start_messaging -f install_dir/etc/messaging.conf.dom2
```

On each managed node, perform steps 8 and 9 from the preceding list, using the same values.

**Note:** Make sure the environment variables `BEA_LC_IPCKEY` and `BEA_LC_CONF_SERVICE` are set to correct values before running the `start_messaging` command on the managed nodes as well as the central host. Also, use the `-i inifile` parameter for all the database commands, such as `lc_user_list`. For more information on database commands, refer to Chapter 3, “Configuring the Database for Use with Log Central.”

# 3 Configuring the Database for Use with Log Central

Log Central has several utilities that configure the database. Use these utilities to perform the following tasks:

- ◆ Managing Your Database
- ◆ Categorizing Messages by Resource
- ◆ Deleting a Subsystem Entry
- ◆ Managing Log Central User Entries

## Managing Your Database

You can manage your database by creating and deleting database schemas.

## Creating a Database Schema

Use the `lc_create_schema` application to create the database table definitions used by Log Central to store message data. These tables are shown in detail in Appendix E, “Database Schema.” The initialization file `msg_processor.ini` contains information about connecting to the database. This file is described in detail in Appendix F, “Initialization File.”

To run this application, simply type `lc_create_schema` on the Central Host.

The syntax of the command follows:

```
lc_create_schema [-infile inifilename]
```

where:

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

If an error occurs, or the script called by the application aborts, run the `lc_drop_schema` application to clean up files that may have been created, correct the problem, and recreate the schema.

## Dropping a Database Schema

Use the `lc_drop_schema` application to drop the Log Central database schema. You might want to do this because the script called by the `lc_create_schema` application aborted. If so, ignore all error messages generated by `lc_drop_schema`, since it may try to drop tables and synonyms that were not yet created. You might also run the `lc_drop_schema` application before reinstalling Log Central.

The syntax of the command follows:

```
lc_drop_schema [-infile inifilename]
```



where:

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

## Categorizing Messages by Resource

You might wish to partition the Log Central database, separating various messages (such as NT events, Oracle messages, BEA TUXEDO messages, and so on) into different categories. These categories represent a resource that generates messages; these resources are known as *subsystems*. Every message has two unique attributes, message ID and subsystem. You can create, add, or drop subsystem entries, and delete subsystems within Log Central, using the `subsystem_create` and `subsystem_delete` commands.

## Creating Subsystem Entries in the Database

Use the `subsystem_create` application to create subsystem entries in the Log Central database. The syntax of the command follows:

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

where:

*subsystem\_name*

Specifies the subsystem name. This is one or more strings, each of up to eight characters. Each string must be entirely in upper case, and it must be unique in the Log Central database. You can supply a list of subsystem names.

*subsystem\_description*

Gives a short description of the subsystem. This is a string of up to 40 characters. If it contains more than one word, enclose the entire string in double quotes.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

An example follows:

```
subsystem_create -s ORACLE -d "Oracle Database Alert Log"
```

**Note:** If you wish only to modify the subsystem description, use the message definition editor. Modifying the subsystem description is described in Chapter 9, “Using the Log Central Console.”

## Deleting a Subsystem Entry

Use the `subsystem_delete` application to delete a subsystem entry from the Log Central database from the command line. All the message *definitions* for the specified subsystem are also deleted; however, the Log Central *messages* for the specified subsystem not deleted. The syntax of the command follows:

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

where:

*subsystem\_name*

Specifies the subsystem name. You can supply a list of subsystem names.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

An example follows:

```
subsystem_delete -s ORACLE
```

# Managing Log Central User Entries

Log Central provides a number of commands for creating and deleting users, modifying user password, and listing available users found in the Log Central database.

## Add a User

Use the `lc_user_create` application to create a new user in the Log Central database. The syntax of the command follows:

```
lc_user_create -u username -p password [-infile inifilename]
```

where:

*username*

Specifies the name of the user to create, which can be up to 10 characters, and contain any alphanumeric character, as well as a hyphen or underscore.

*password*

Specifies the user password, which can be up to 10 characters, and contain any printable ASCII character.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

An example follows:

```
lc_user_create -u simpson -p simps0n
```

## Delete a User

Use the `lc_user_delete` application to delete a user from the Log Central database. The syntax of the command follows:

```
lc_user_delete -u username -p password [-infile inifilename]
```

where:

*username*

Specifies the name of the user to delete.

*password*

Specifies the password.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

An example follows:

```
lc_user_delete -u simpson -p simpson
```

## Modify a Password

Use the `lc_user_modify` application to modify a user password in the Log Central database. The syntax of the command follows:

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

where:

*username*

Specifies the name of the user to modify.

*oldpassword*

Specifies the current password.

*newpassword*

Specifies the new user password, which can be up to 10 characters, and contain any printable ASCII character.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

An example follows:

```
lc_user_modify -u simpson -p simps0n -n slmps0n
```

## List Users

You can use the `lc_user_list` application to list all the users in the Log Central database. The syntax of the command follows:

```
lc_user_list [-infile inifilename]
```

where:

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”



# 4 Integrating Logs into Log Central

The Log Monitor program integrates logs into Log Central. Log Monitor does the following:

- ◆ Monitors application logs for incoming messages.
- ◆ Selects which incoming log messages are to be forwarded to the Log Central database.
- ◆ Maps the attributes in log messages into the internal Log Central log message format.

A number of predefined mappings of log formats are provided with Log Monitor. For information on how to start Log Monitor using a predefined mapping, refer to Chapter 8, “Starting and Stopping Log Central.” This chapter describes how to construct mappings for additional log files that you want to monitor. Consult Appendix A, “Message Format,” for information about the Log Central log message format.

There are two ways to pass a mapping to Log Monitor when it is started:

- ◆ As a series of options in a Log Monitor configuration file
- ◆ As options passed on the command line

This chapter describes the configuration file options. Passing the options on the command line, and the syntax used in starting Log Monitor, are discussed in Chapter 8, “Starting and Stopping Log Central.”

# Log Monitor Configuration File Options

A Log Monitor configuration file is simply a list of up to 20 different Log Monitor filters, each on a separate line. Each filter consists of a list of Log Monitor options.

Filters can be used to specify which messages to drop. For example, if you use the `-p` pattern option in a filter, a log message that satisfy this filter is then forwarded by Log Monitor. If this is the only filter in the file, any message not matching the pattern specified by the `-p` option will be discarded.

Alternatively, you can use the `-x` option to specify particular messages that you do not want selected. To select which message to ignore, the `-x` option also uses a pattern to determine a match. Using patterns with the `-x` and `-p` options is discussed below under “Using the `-p` and `-x` Options.”

Log Monitor tests each incoming message against each filter sequentially. A log message that fails to be selected by one filter in the configuration file may be selected by one of the other filters. A log message is forwarded by the Log Monitor to the Log Central database if it is selected by at least one of the filters in the configuration file.

You may want to use multiple filters to provide different mappings for different message types.

If you use a configuration file to instruct Log Monitor how to map incoming log messages, the name of the configuration file is passed to Log Monitor in the `-f` option when it is started.

Table 4-1 summarizes the options that are available for defining a mapping in a configuration file.



**Table 4-1 Log Monitor Configuration File Options**

Argument	Description
<code>-b body</code>	<i>body</i> is a string that contains the body of the message.
<code>-D date %f"format"</code>	This option specifies the date format to use in the message. See “Specifying the Date Format (%f).”
<code>-d msgid</code>	This option specifies the message ID to use in the message. The default is 1000.
<code>-I processID</code>	This option specifies the process ID to use in the message. The default is the process ID of the Log Monitor daemon process.
<code>-M log_level</code>	<p>This option specifies the logging level to use in the message. This is a one-character string. The allowed values and their interpretation are as follows:</p> <p>N—Normal V—Verbose D—Debug S—Special</p>
<code>-m subsystem</code>	This option specifies the module subsystem name to use in the message. The default is none.
<code>-n function</code>	This option specifies the function name to use in the message. The default is none.
<code>-o hostname</code>	This option specifies the host name to use in the message. The default is the machine on which Log Monitor is running.
<code>-p pattern</code>	This option instructs the program to forward only messages that match <i>pattern</i> , which may use special metacharacters. Use the metacharacters defined under “Using the -p and -x Options.”

**Table 4-1 Log Monitor Configuration File Options**

Argument	Description
<code>-S</code>	<p>This option specifies one or more separators to be used to calculate field values in the input message file for corresponding <code>%F</code> specifiers on a command line. (The <code>%F</code> format symbol is described in Table 4-2.)</p> <p>If more than one separator is specified, all are used to count the fields. If a message starts with a separator, the text between the first and the second separator is counted as field number 1. Fields are numbered starting with 1 (not 0). If a message does not start with a separator, the first field consists of the text up to the first separator.</p> <p>For examples, see “Multiple Separators with the <code>-S</code> Option.”</p>
<code>-T transactionID</code>	<p>This option specifies the transaction ID to use in the message. The default is 0.</p>
<code>-u userID</code>	<p>This option specifies the user ID to use in the message. The default is the current user (that is, the owner of the <code>log_monitor</code> process).</p>
<code>-x pattern</code>	<p>This option instructs the program to not forward messages that match <i>pattern</i>. To define a pattern, you can use the metacharacters defined under “Using the <code>-p</code> and <code>-x</code> Options.”</p>

Table 4-2 Format Symbols

Character	Description	Examples
%F	Used with an integer to specify a particular field. Must be accompanied by -S (from Table 4-1).  The -S option specifies the character used to separate fields in the message.	-m %F1 -S ' ' '  This selects the first field from the incoming message and uses it as the subsystem name in the corresponding Log Central message to be generated. The field separator is the vertical bar.
%C	Specifies the starting character position for the value. Must be followed by the L or S symbol (not both) to terminate the value.  The L symbol specifies an absolute length in characters of the value.  The S symbol specifies that the following character is to be a separator.	-u %C10L4  This selects the 10th, 11th, 12th, and 13th characters from the incoming message to use as the user ID in the corresponding Log Monitor message to be generated.  -u %C10S   This selects the string that starts at the 10th character and ends immediately before the next   character.
%V	Specifies user-defined values within formats.	-n %C3L5%F11%V"This option indicates a MINOR FUNCTION" -S:  This creates a function name entry composed of the five-character string starting at character 3, field 11, and the string This option indicates a MINOR FUNCTION. The field separator is the colon.
%f	Specifies the format for the date value. This option is used with the -D option only.  All values of %f are detailed in Table 4-4.	-D %F1%f"format"  This specifies that the date value in field 1 is in the format as specified by the format string ( <i>format</i> ).  Double quotes are required only if this option is being given on the command line (or if <i>format</i> contains one or more embedded spaces).

## Specifying Option Values

You can specify how the value for an option is determined in two ways: by literal value and by format. The methods can be mixed.

To specify the option by literal value, follow the option with the specific value. For example:

```
-u KONG
-b "This is the message body."
-m "Major function"
-d 1234
```

If you specify both a literal value and a format, the literal must start with the characters %v, which is explained in Table 4-2.

To specify the value by format, use format symbols to extract the value from the application log message. Examples appear in Table 4-2 under %C and %v.

## Example of Using Log Monitor with a Configuration File

The following is an example of a command used to start Log Monitor with options in a configuration file.

```
log_monitor -f forward_options -i /home/demo/demo.mul_fld -t 0
```

The configuration file is named `forward_options`. The contents of `forward_options` are:

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

Each line pertains to one filter or mapping for the `/home/demo/demo.mul_fld` log file. Log Monitor works on these filters sequentially. Log Monitor picks one message, applies the three filters one after the other, then moves to the next message in the input file.

The separator specification following the `-S` option indicates that either `|` or `!` is considered to be a separator. Whichever one the parser encounters signals the start of a new field. For an explanation of the `-S` option, refer to “Multiple Separators with the `-S` Option.”

## Multiple Separators with the -S Option

When you use the `-s` option, you can specify multiple separator characters. The following example shows how fields are numbered in such situations.

If the following is the incoming message:

```
abcd^xys^b|bbbb^
```

Specifying `-s^|` for the separator results in the following.

Field Number	Contents
1	abcd
2	xys
3	b
4	bbbb

Specifying `-s^` for the separator results in the following.

Field Number	Contents
1	abcd
2	xys
3	b bbbb

Separators at the beginning of a message are ignored. For example, the fields would be exactly the same as the previous table if the incoming message were:

```
|abcd^xys^b|bbbb^
```

# Using the -p and -x Options

The `-p` option specifies a pattern that is used to select messages for forwarding. If a message does not match the pattern, it is not selected for forwarding by that filter.

The pattern may simply be a string. For example, if you use:

```
-p su:
```

then any message in which `su:` occurs will be selected. You can also use metacharacters to select a range of values. For example, if you want to select messages with a year value from 97 to 99, you could use:

```
9[7-9]
```

to specify such a range.

The `-x` option specifies a pattern that is used to select messages to be discarded. If a message matches the pattern, it is not selected by that filter. (However, that message might still be selected for forwarding by another filter if you are using multiple filters in the same configuration file.)

Table 4-3 lists the metacharacters you can use with the `-p` and `-x` options.

**Table 4-3 Metacharacters**

Expression	Description
<code>?</code>	Matches any single character except a newline character.
<code>%</code>	Matches the beginning of the line. For example,  <code>%abc</code> matches a string only if the letters <code>abc</code> occur as the first three characters of a line. The <code>%</code> symbol does not have its special metacharacter role if it is not at the beginning of a line.
<code>\$</code>	Matches the end of a line. For example,  <code>xyz\$</code> matches a string only if the letters <code>xyz</code> occur as the last three characters on the line. The <code>\$</code> symbol does not have its special metacharacter role if it is not at the end of a line.

**Table 4-3 Metacharacters**

Expression	Description
<code>@c</code>	Escapes the following character ( <i>c</i> ). When followed by any metacharacter, the expression matches the metacharacter itself. For example, <code>@%</code> matches a percent sign (which otherwise would be interpreted as part of an expression specified as starting at the beginning of a line).  Do not confuse this with the backslash used on the command line to escape characters that have special meaning to the command interpreter.
<code>*</code>	Indicates multiple 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 <code>aaabaa</code> , the expression <code>a*</code> would match <code>aaa</code> .
<code>[string]</code>	Indicates a group of characters. A nonempty string of characters enclosed in square brackets matches any one character in the string. If the first character is a caret ( <code>^</code> ), the regular expression matches any character <i>except</i> a newline character and the remaining characters in the string. Use a hyphen to indicate a range of consecutive ASCII characters, such as <code>[0-9]</code> .

Here are examples of metacharacter usage with `-x` and `-p`:

`[aeiou]`

Matches any single character that is a vowel.

`[^a-zA-Z0-9]`

Matches any nonalphanumeric character.

`-x [a-zA-Z][a-zA-Z]*$`

Do not forward any messages containing lines that end in words. The specification is to match an entire word, that is one containing one or more alphabetic characters.

`-p (?*)`

Forward any messages with lines containing parentheses.

`-x %gobbledegook$`

Ignores any message with a line that consists solely of `gobbledegook`.

# Field Lengths

If the length of a string-valued field in the input file goes beyond its maximum, the value is truncated. For example, if an input message contains the user ID Administrator, it would be truncated to Administ.

The maximum lengths of the fields are the following.

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



## Specifying the Date Format (%f)

When specifying the date by format, use the date format detailed in Table 4-4.

**Table 4-4 Date Formats**

Format	Explanation
%%	A literal percent sign
%a	Abbreviated weekday name (for example, Sun)
%A	Full weekday name (for example, Sunday)
%b	Abbreviated month name of the locale (for example, Jan)
%B	Full month name of the locale (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)
%m	Month number (1-12; leading zeroes are permitted but not required)
%M	Minute (0-59; leading zeroes are permitted but not required)
%p	Locale's 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 within century (0-99; leading zeroes are permitted but not required)
%Y	Year, including century (for example, 1998)

## Filtering a System Log

This section gives an example of filtering a system log. The following file contains messages from the UNIX system log (/var/log/syslog):

```
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
```

Our configuration file (conf\_file) contains the following lines:

```
-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-
```

To filter the UNIX system log, run the following command:

```
log_monitor -i /var/log/syslog -f /home/emilie/conf_file
```

(For log\_monitor command options, refer to Appendix 8, “Starting and Stopping Log Central.”)

This produces the following Log Central messages:

```
|N|May 16 13:51:11
1998|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 1998|AUTH|124|eclipse|11593|emilie|su:|0|1!'su
webuild' failed for emilie on /dev/ttyp4

|V|May 17 13:54:28
1998|NFS|123|eclipse|11593|emilie|vmunix:|0|1!NFS write error: on
host iseult remote file system full

|N|May 17 14:40:42
1998|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 1998|AUTH|124|eclipse|11593|emilie|su:|0|1!'su
root' succeeded for emilie on /dev/ttyp0
```

The following table shows how the mapping in the second line in `conf_file` produced the second Log Central message shown in the preceding. Two generated lines of output (the second and fifth messages in the preceding) were produced as a result of that mapping.

Input	Format Specification	Meaning	Output
su:	-p su:	Each message containing the specified string (su: in this case) is mapped according to the specifications in the second line of <code>conf_file</code> .	Log Central sample messages 2 and 5 (shown preceding this table).
None.	None provided.	Reporting mode; uses the default, <code>LM_NORMAL</code> , which produces an N in the Log Central file.	N
May 17 14:40:42	-D "%F1%V %F2%V %F3%f%h %d %T" -S " "	Time stamp; constructed from fields 1, 2, and 3.	May 17 14:40:42 1998
Field not present in input.	-m AUTH	Subsystem name; specified by literal value.	AUTH
Field not present in input.	-d 124	Message ID; specified by literal value.	124
eclipse	-o %F4	Host name; taken from field 4.	eclipse

## 4 INTEGRATING LOGS INTO LOG CENTRAL

Input	Format Specification	Meaning	Output
Field not present in input.	None.	Process ID; uses default (PID of the Log Monitor process), since there is no <code>-I</code> specification.	11593
Field not present in input.	<code>-u emilie</code>	User name; specified by literal value.	emilie
su:	<code>-n %F5</code>	Function name; extracted from field number 5.	su:
Field not present in input.	None.	Transaction ID; uses default (0).	0
Field not present in input.	None.	Reserved.	1!
'su webuild' failed for emilie on /dev/ttyp4	<code>-b %F6-</code>	Message body; constructed from field 6 until the end of line.	'su webuild' failed for emilie on /dev/ttyp4

**Note:** 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.

Where fields are specified, the field numbers are calculated by using the separator specified with the `-s` option, which is a blank in the preceding example (the second line in `conf_file`).

## Specifying Date Format

To specify the date format, precede the data format with %f.

If the input file contains lines like the following:

```
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
```

To map these dates, you could use the following:

```
-D "%F1%V %F2%V %F3%f%h %d %T" -S " "
```

If the input file contains a line like the following:

```
eclipse|su:|12/12/99 09:20|'su root' succeeded for emilie on  
/dev/tty0
```

To map the date, you could use the following:

```
-D %F3%f"%D %H:%M" -S|
```

## Converting Input Dates

The following rules are applied for converting the input specification into the internal format:

- ◆ If only the weekday is given (that is, without specifying a day of the month), today is assumed if the given day is equal to the current day. Otherwise, the corresponding day from the next week is assumed.
- ◆ If only the month is given (that is, without specifying a year), the current month is assumed if the given month is equal to the current month. Otherwise, the corresponding month from the next year is assumed. The first day of the month is assumed if no day is given.
- ◆ If only the time is given (that is, without specifying a date), today is assumed if the given hour is greater than the current hour. Otherwise, the corresponding time from tomorrow is assumed. If no hour, minute, and second are given, the current hour, minute, and second are assumed.

## 4 INTEGRATING LOGS INTO LOG CENTRAL

---

The following examples illustrate these rules. Assume that the current date is Tue Sep 22 12:19:47 PDT 1998.

Input	Line in Template	Date
Tue	%a	Sep 22 12:19:47 PDT 1998
Mon	%a	Sep 28 12:19:47 PDT 1998
Fri	%a	Sep 26 12:19:47 PDT 1998
September	%B	Sep 1 12:19:47 PDT 1998
January	%B	Jan 1 12:19:47 PST 1999
December	%B	Dec 1 12:19:47 PST 1998
Sep Tue	%b %a	Sep 1 12:19:47 PDT 1998
Jan Sat	%b %a	Jan 2 12:19:47 PST 1999
Dec Tue	%b %a	Dec 1 12:19:47 PST 1998
Jan Fri 2003	%b %a %Y	Jan 3 12:19:47 PST 2003
Fri 9	%a %H	Sep 26 09:00:00 PDT 1998
Feb 10:30	%b %H:%S	Feb 1 10:00:30 PST 1999
10:30	%H:%M	Sep 23 10:30:00 PDT 1998
13:30	%H:%M	Sep 22 13:30:00 PDT 1998

# 5 Creating and Loading Message Definitions

This chapter describes the following:

- ◆ Message Definitions
- ◆ Getting Message Definitions into the Log Central Database
- ◆ Using Other Message Definition Commands

## Message Definitions

Applications generate messages in many different formats. Log Central stores messages in its database in one format. It also stores *message definitions*. Message definitions provide static information about messages that may be generated by various applications. Such information might include recommendations for how to respond to specific errors. Message definitions can be viewed within the Log Central Console with the Message Details window of the Message Browser. For more information about the Message Browser, refer to Chapter 9, “Using the Log Central Console.”

Message definitions provide information about probable causes or suggest corrective actions in response to received messages. Two fields in the Log Central message format—Description and Recommendation—are provided for this purpose. You can access this information from the Log Central Console by invoking the Message Browser Message Details window, as described in Chapter 9, “Using the Log Central

Console,” or by exporting message definitions to a file, as described in this chapter in “Exporting Message Definitions.” This information is stored in the message type definitions in the Log Central database.

Message definitions are provided out-of-the-box for BEA TUXEDO applications, Oracle Host Log, and NT eventlog messages. Utilities, which are described in this chapter, are provided to add message definitions for other resources you wish to manage. The Message Definition Editor of the Log Central Console can also be used to update these fields. This facility allows users to update probable cause or recommended action fields in light of their experience.

## Getting Message Definitions into the Log Central Database

To get message definitions into the Log Central database, you perform the following two steps:

1. Create a Message Definition File
2. Load the Message Definition File

You can also modify or create message definitions one at a time through the GUI. How to do this is described in Chapter 9, “Using the Log Central Console.”

### Create a Message Definition File

To create a message definition file, perform one of the following two actions:

- ◆ Construct your own file, following the structure and rules of the template (*install\_dir/etc/msgdef.template*) supplied with Log Central.
- ◆ Copy the template and then edit it to your specifications.

The remainder of this section provides details.



## Description of Message Definition File

A message definition file contains one or more message definitions, each of which consists of up to 10 fields. Two of those fields, `SUBSYSTEM` and `SEVERITY`, must appear in the first message definition, and can optionally appear in succeeding definitions. If they are not specified in any definition after the first, their values are inherited from the last definition in which they were specified. Severity is an important attribute of a message. The severity of a message is a rating used to represent the importance or impact of an event. For example, a message that indicates high usage of a print spooler reports a less severe event than a message telling you that an application server has crashed. The Log Central Central Collector assigns a severity to messages as it saves the message in the Log Central database or to generate SNMP traps. When the Log Central Console displays messages in its Message Browser, the messages are typically colored depending on their severity. If you want the same subsystem and severity to apply to all message definitions, you can specify these values only in the first definition. Typically, though, you would assign a different severity to each message definition, and thus would repeat this field for each message definition. Alternatively, you could group message definitions by severity, so as to preclude having to repeat the same severity for each definition in a group.

The subsystem and message ID are used to correlate incoming messages in the Log Central database with their corresponding message definitions. Thus, when a message with a particular subsystem and message ID arrives, you can display its corresponding message definition.

Two of the fields, `MESSAGE_ID` and `SUMMARY`, must appear in every message definition. The message ID uniquely identifies a message definition, while the message summary gives it contextual identity.

Other fields are optional. If a particular field appears in one message definition, but not in succeeding definitions, the value of that field is inherited by message definitions that follow. Each message definition is enclosed within a pair of braces (`{ }`).

The message definitions are stored in an ASCII text file. You can either construct your own file, following the structure and rules of the template supplied with Log Central, or copy the template and then edit it to your specifications. The template is found in `install_dir/etc/msgdef.template`. If you use the template, you must edit all constructs following equals signs because the constructs specified in the template file are just descriptive placeholders.

## Fields of a Message Definition File

The fields of a message definition are shown in the following table in the order in which they appear in the message definition file.

Field Name	Format	Description
SUBSYSTEM	String field, single line, maximum of 8 characters.	Subsystem name (for example, TUXEDO, syslog), which must appear in the first message definition.
MESSAGE_ID	Integer, appearing on a single line, in the range 1-99999.	The message ID, which must appear in every message definition, which must be unique within a subsystem.
SUMMARY	String field, single line, maximum of 40 characters.	Summary of the DESCRIPTION field, which must appear in every message definition.
SEVERITY	One of the following strings: Informational, Warning, Minor, Major, Critical	The severity level of the message.
DESCRIPTION	String field, on one or more lines. The total maximum of this field plus the RECOMMENDATION field must not exceed 2 Gb.	The text to be associated with the message.
RECOMMENDATION	String field, on one or more lines. The total maximum of this field plus the DESCRIPTION field must not exceed 2 Gb.	A recommendation of what to do when encountering this message.
EXECUTE_ON_UPLOAD	String field, single line, maximum of 40 characters.	The path to a script or executable file to invoke when the Central Collector loads the specified message into the Log Central database.

Field Name	Format	Description
TRAP_ID	Integer, appearing on a single line, in the range 1-99999.	The specific SNMP trap ID to issue upon encountering this message.
TRAP_ENABLED	A string, one of Yes or No.	A flag indicating whether the SNMP trap is to be enabled. Default is No.
AUTO_ACKNOWLEDGE	A string, one of Yes or No.	A flag indicating whether the message is to be automatically acknowledged. Default is No.

## Example

A sample message definition might look like the one following this paragraph. This definition includes all of the possible fields. Its subsystem is `TUXEDO`, its `MESSAGE_ID=1206`, its severity is `Critical`. The summary, description, and recommendation are as shown. When the Central Collector loads a message of ID 1206 and subsystem `TUXEDO`, the file `sendalert.exe` is executed. This could, for example, send an alert to the system administrator. Whenever a message matching this `MESSAGE_ID` and subsystem appears, Log Central issues an SNMP `TRAP_ID=47`. The message is also automatically marked as acknowledged.

```
{
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
}
```

## Load the Message Definition File

Once you have constructed your own message definition file, you load it into the Log Central database by entering the following command at the command-line prompt:

```
msgdef_import [-f filename] [-infile inifilename]
```

where:

*filename*

Specifies the complete file name from which the message definitions are to be imported into the Log Central database. The contents of this file must follow a specified format. If no file name is given, text is accepted from the standard input of the terminal.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

# Using Other Message Definition Commands

There are also command-line utilities that enable you to export message definitions from the Log Central database to an ASCII text file and delete selected message definitions.

## Exporting Message Definitions

You can export message definitions from the database to an ASCII text file with the `msgdef_export` command at the command-line prompt. The exported data is written to a file in the format described in “Create a Message Definition File.” To export message definitions, use the following command at the command-line prompt:

```
msgdef_export [-f filename] [-s subsystem_name  
[-s subsystem_name...]] [-infile inifilename]  
  
msgdef_import [-f filename] [-infile inifilename]
```

where:

*filename*

Specifies the complete file name to which the message definitions are to be exported. If no file name is given, the message definitions are displayed on the standard output of the terminal.

*subsystem\_name*

Specifies the subsystem name for which message definitions will be exported. If no subsystem is given, the message definitions for all subsystems are exported. You can supply a list of subsystem names.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

## Deleting Message Definitions

You can delete the message definitions specified in the message definition file from the Log Central database with the `msgdef_delete` and `subsystem_delete` commands at the command-line prompt. If you want to delete a group of message definitions you loaded previously, you could use the same message definition file you used with the `msgdef_import` command. If you want to delete a group of message definitions that belong to a specific subsystem, use the `subsystem_delete` command.

### How to Delete a List of Message Definitions

You can delete a list of message definitions by using the following command:

```
msgdef_delete [-f filename] [-infile inifilename]
```

where:

*filename*

Specifies the complete file name that contains a list of message definitions to be deleted. The contents of this file must follow a specified format, as described in “Create a Message Definition File.” Only the subsystem name and message ID are used for the deletion operation. If no file name is given, text is accepted from the standard input of the terminal.

*inifilename*

Specifies the complete file name of the initialization file. If no name is given, the name `msg_processor.ini` is used. The initialization file is described in detail in Appendix F, “Initialization File.”

You might use this command to delete the message definitions you just loaded with `msgdef_import`.

### How to Delete Message Definitions Associated with a Subsystem

You can delete all the message definitions associated with a particular subsystem, as well as the subsystem name entry. To do so, use the following command:

```
subsystem_delete -s subsystemname [subsystem_name]
```

where:

*filename*

Specifies the complete file name to which the message definitions are to be exported. If no file name is given, the message definitions are displayed on the standard output of the terminal.

*subsystem\_name*

Specifies the subsystem name for which to delete all message definitions. You can supply a list of subsystem names. Only the subsystem name and message ID are used for the deletion operation.





# 6 Host and Filter Configuration

When Log Central starts, it reads a configuration file, `messaging.conf`, to determine central host configuration and agent filtering. Some of the configuration attributes in this file are set up for you when you run the Host Configuration Utility (`lc_config`), as described in Chapter 2, “Getting Started.” This chapter describes optional additional fields in the configuration file that you may want to modify. Topics discussed include:

- ◆ Default Configuration
- ◆ Specifying a Backup Central Collector
- ◆ Specifying Nondefault Global Parameters
- ◆ Assigning Filters to Agents
- ◆ Defining Agent Filters

Log Central uses a single configuration file located on the central host. The configuration file is an ASCII text file which you can modify with your favorite text editor. The default location for the configuration file is:

UNIX systems

`install_dir/etc/messaging.conf`

Windows NT

`C:\install_dir\etc\messaging.conf`

where:

`install_dir`

Is the directory where Log Central was installed.

# Default Configuration

When you complete the initial setup using the Host Configuration Utility (`lc_config`), your configuration is specified by an `LC_GLOBAL` entry. This entry provides the name of the central host and the directory and prefix for intermediate message files used by the Message Receiver to pass log messages to the Message Processor for inclusion in the relational database. For example:

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

At the minimum, the configuration file must contain an `LC_GLOBAL` clause, in which the keywords `CENTRAL_HOST` and `LOGPREFIX` must appear. The `LC_GLOBAL` clause configures parameters for all nodes. The configuration file can contain other constructs, but they are optional.

The Log Central configuration file consists of commented sections that contain all the permitted constructs you might need. After initially running the Host Configuration utility, only the `CENTRAL_HOST` and `LOGPREFIX` fields in the `LC_GLOBAL` entry are not commented out. (In the configuration file the pound sign (#) indicates the start of a comment.) However, you should look at the commented sections to get an idea of the constructs that you can use in the configuration file to further customize your configuration. Any modifications you make to the configuration file only take effect after the `start_messaging` command is used to start up the Log Central components.

## Where to Put Intermediate Files

When the intermediate log files (specified by `LOGPREFIX`) reside on a remote file system, performance of the Log Manager is adversely affected.

We recommend using log files on the local file system if the log message traffic is high.

# Specifying a Backup Central Collector

To specify a backup Central Collector, you need to add the following two lines to the `LC_GLOBAL` entry in the configuration file:

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

For example, you might have an `LC_GLOBAL` entry that looks like this:

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

To use a backup Central Collector, the name of the backup host must be specified when starting each of the log agents. Both the primary and backup Central Collectors must be started before the agents are started.

For best performance, we recommend specifying local file systems for the `LOGPREFIX`, `BACKUP_LOGPREFIX`, and `ESCALATION_DIR` keywords in the `LC_GLOBAL` clause.

## Host Name Usage

The host names used in the configuration file (`CENTRAL_HOST`, `BACKUP_HOST`, and `HOSTNAME`) are the same as the value returned by the `hostname` command on the specified systems. For example, on a particular host, `hostname` returns `star.abc.com`, then the identifier for that host is `star.abc.com` and not `star`.

# Specifying Nondefault Global Parameters

The `LC_GLOBAL` entry in the configuration file can also be used to specify nondefault values for the following:

- ◆ IPC key

The default IPC key value is `0xeeee0000`. Use the `IPCKEY` keyword in the `LC_GLOBAL` entry to define a nondefault IPC key value.

`IPCKEY` can take any numeric value. Use `0x` to indicate a hexadecimal value.

- ◆ The communication service between the Message Sender and Message Receiver

The default service name used for communication between the Message Sender (`msg_sender`) processes and the Message Receiver (`msg_receiver`) process running on the central host or on the backup host is `lc_talk`. Use the `TALK_SERVICE` keyword in the `LC_GLOBAL` entry to specify a nondefault service name.

- ◆ The location for the message processor initialization file (`msg_processor.ini`)

`msg_processor.ini` is the initialization file used by the `msg_processor` process. The default location of this file is `install_dir/etc/msg_processor.ini`, where `install_dir` is the directory where Log Central was installed on the central host. To specify a different location for the Message Processor initialization file, use the `INIFILE` keyword in the `LC_GLOBAL` entry.

- ◆ Temporary log file directory for data collection agent on managed nodes

When a data collection agent is unable to forward messages to a Central Collector, it writes log messages into a temporary log file. When the Central Collector becomes available, the data collection agent automatically recovers the contents of the file and forwards the messages to the Central Collector. The default directory for these files is `/tmp` on UNIX, and `C:\tmp` on Windows NT. Use the `ESCALATION_DIR` keyword in the `LC_GLOBAL` entry to specify a different directory for these log files. This directory name is global since it is the same on all nodes that connect to this Central Collector.

All of the valid keywords that you can use in the `LC_GLOBAL` entry are described in Table 6-1.

**Table 6-1 LC\_GLOBAL Keywords**

<b>Keyword</b>	<b>Data Type</b>	<b>Description</b>
CENTRAL_HOST	<i>string</i>	The host name of the central host, which is a mandatory field.
LOGPREFIX	<i>string</i>	The base name of the intermediate log message files used by the <code>msg_receiver</code> process running on the central host. This is a mandatory field.
BACKUP_HOST	<i>string</i>	The host name of the machine where the backup Central Collector runs. This field is optional. If this field is not specified, no backup central host configuration is made.
BACKUP_LOGPREFIX	<i>string</i>	The base name of the intermediate log message files used by the <code>msg_receiver</code> process running on the backup central host. This field is optional unless the <code>BACKUP_HOST</code> keyword has been specified, in which case it is mandatory. There is no default for this field.
TALK_SERVICE	<i>string</i>	The service name used for communication between the <code>msg_sender</code> processes and the <code>msg_receiver</code> process running on the central host or on the backup host. This is an optional field. The default value of this field is <code>lc_talk</code> .
IPCKEY	<i>integer</i>	The IPC key for the Log Central subsystem. This is an optional field, unless you have more than one Log Central system. For details on how to handle this situation, refer to Appendix B, “Environment Variables.” The default value of this field is <code>0xeeee0000</code> . <code>IPCKEY</code> can take any numeric value. Use <code>0x</code> to indicate a hexadecimal value.
INIFILE	<i>string</i>	The initialization file used by the <code>msg_processor</code> process. This is an optional field. The default value of this field is <code>install_dir/etc/msg_processor.ini</code> .
ESCALATION_DIR	<i>string</i>	The directory for storing temporary log files on managed nodes. The name of the temporary log file directory is the same for all of the Log Central hosts. (It can be a local or remote file system, but the name should be same.) This is an optional field. The default value of this field is <code>/tmp</code> on UNIX ( <code>C:\tmp</code> on NT).
FILTER	<i>string</i>	The filter identifier, if any filtering is required. (This option can be repeated to specify multiple filters.) This is an optional field. There is no default for this field. If no <code>FILTER</code> entry is specified in the <code>LC_GLOBAL</code> entry, no global filter is present.

# Assigning Filters to Agents

Agent filters can be used by the data collection agents to do the following:

- ◆ Drop specified messages (that is, fail to forward them to the Central Collector)
- ◆ Execute a program or script when a specified log message occurs
- ◆ Send an SNMP trap notification to an enterprise management system

Defining filters is described under “Defining Agent Filters.”

Use of agent filters is optional. By default, no agent filters are defined.

Filters are invoked in either the `LC_GLOBAL` entry or the `MANAGED_NODE` entries in the Log Central configuration file. In either case, a filter is invoked if a statement occurs in the entry that has this form:

```
FILTER = "filtername"
```

where *filtername* cannot be longer than eight characters.

For each named filter that is invoked in this way in the configuration file, there must be a `DEFINE_FILTER` entry in the configuration file that defines the named filter.

An agent filter can be global or it can be assigned to a specific machine (managed node). An agent filter is global if it is invoked in the `LC_GLOBAL` entry in the configuration file. For example:

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

`BankAppTrap` might be a filter that generates SNMP traps when certain log messages occur, for example. Of course, if you use a `FILTER` statement, this presupposes that a `DEFINE_FILTER` entry occurs in the configuration file, defining the invoked filter—`BankAppTrap` in this case. `DEFINE_FILTER` entries are described in “Defining Agent Filters.”

## Using a **MANAGED\_NODE** Entry

If an agent filter is global, then by default all agents use that filter. However, you can also:

- ◆ Turn off the global filter for a particular managed node
- ◆ Assign a particular filter to a particular managed node

To do either of these things requires a **MANAGED\_NODE** entry in the configuration file.

The **MANAGED\_NODE** entry is simply a list of assignments defining the filters for a managed node in your Log Central system. A **MANAGED\_NODE** entry is required only if you wish to specify non-global filters. No more than one **MANAGED\_NODE** entry may be specified for each managed host.

If a **MANAGED\_NODE** entry is not specified for a managed node, that node inherits all the global filters (if any), as specified in the **LC\_GLOBAL** entry. The simplest approach is to use global filters for situations that apply to the greatest number of hosts, and specify any exceptions within the **MANAGED\_NODE** entry.

## Turning Off Global Filters on a Particular Node

If you want to turn off the global filters for a particular managed node, use the **GLOBAL\_FILTER** statement in a **MANAGED\_NODE** entry for that managed node. Global filters are turned off for machine **bigiron** in the following example:

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

## Assigning a Filter to a Particular Node

A `MANAGED_NODE` entry can be used to assign a particular filter to a particular machine. You might do this if you want to define a custom filter for a particular log resource being managed by a single agent. For example:

```
MANAGED_NODE
{
    HOSTNAME      = "peach"
    FILTER        = "F4"
}
```

In this example, filter `F4` is used by the log agent on machine `peach`, in addition to any global filters that may be defined in the `LC_GLOBAL` entry. In the following example, the agent on machine `marmalade` does not use any global filters but applies filters `F2` and `F3` to incoming log messages:

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

Valid keywords for the `MANAGED_NODE` entry are described in Table 6-2.

**Table 6-2 Valid Keywords for the `MANAGED_NODE` Entry**

Keyword	Data Type	Description
<code>HOSTNAME</code>	<i>string</i>	The host name to which this <code>MANAGED_NODE</code> entry applies. This is a mandatory field.
<code>FILTER</code>	<i>string</i>	The filter identifier, if any filtering is required. (This option can be repeated to specify multiple filters.)
<code>GLOBAL_FILTER</code>	<i>string</i>	Specifies whether to include global filters or not (if this keyword is not specified, global filters are included). Valid values are <code>YES</code> and <code>NO</code> .



# Defining Agent Filters

In order to invoke a filter in an `LC_GLOBAL` or `MANAGED_NODE` entry, the filter must of course be defined, in a `DEFINE_FILTER` clause. If you use filters, the `DEFINE_FILTER` clause must appear in the configuration file before the `LC_GLOBAL` clause.

A `DEFINE_FILTER` entry has the following syntax:

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

You can specify one or more actions to be taken when *condition* evaluates to true.

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

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

This entry defines a filter named `DropInfo`. This filter specifies that a message having a message ID of 8 is dropped, that is, it is not sent to the Log Central Central Collector. You might use such a filter to drop messages that you do not wish to monitor. By default, data collection agents send all messages to the Central Collector, that is, the default value of `REMOTE` is `YES`.

Complex conditions can be built up from simple conditions using logical operators. The following logical operators can be used to build up complex conditions.

**Table 6-3 Logical Operators for Use in Defining Conditions**

Syntax	Interpretation
<code>!( condition )</code>	Evaluates to true when <i>condition</i> is false.
<code>( condition1 ) &amp;&amp; ( condition2 )</code>	Evaluates to true if both <i>condition1</i> and <i>condition2</i> are true.
<code>( condition1 )    ( condition2 )</code>	Evaluates to true if either <i>condition1</i> or <i>condition2</i> (or both) is true.

For instance, the earlier `DropInfo` example might not suit your need if you are monitoring two subsystems which both have messages with a message ID of 8. You might want to only drop messages with a message ID of 8 from one of the subsystems. The following filter drops only messages with a message ID of 8 from subsystem `NDB`:

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

Only attributes in the message header, or contents of the message body, can be used to define message filtering because only these components of the message are available to the data collection agents. The attributes contained in the message definition cannot be used for filtering at the local agent because they are stored in the Log Central database and are only accessible to the Central Collector.

## Defining Conditions

The following keywords can be used to define conditions.

**Table 6-4 Keywords for Use in Defining Conditions**

Keyword	Data Type	Message Attribute
PID	Number	Process ID
MSGID	Number	Message ID
SUBSYSTEM	String	Subsystem name
LOG_LEVEL	String (one of N, V, D, or S)	Logging level
HOST	String	Host name
FUNCTION	String	Name of the internal function that issued the message. This can be up to 40 characters in length.
TXKEY	String	Transaction key. This string can be up to 20 characters in length.
USER	String	User ID
MSGBODY	String	String to test for a match in the body of the message

**Note:** String values must be enclosed in double quotes.

Table 6-5 describes the relations that can be used in defining conditions. Note that the symbols have a somewhat different interpretation for string arguments than for numeric arguments. Perhaps the most useful relational operator for strings is `>=`.

**Table 6-5 Relations for Use in 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

## Defining Actions

You can specify one or more actions to be taken by a data collection agent when a condition evaluates to true. Three types of action statement are possible:

- ◆ Dropping a Message
- ◆ Executing a Command
- ◆ Sending an SNMP Trap Notification

### Dropping a Message

You can instruct the data collection agent not to send the message to the central host: To do so, use an action statement of the following form:

```
REMOTE = "NO"
```

## Executing a Command

You can instruct the data collection agent to execute a program or invoke a shell script or batch file. To do so, use an action statement of the following form:

```
COMMAND = "executable_path [arguments]"
```

*executable\_path*

The name of the program or script, including the full path.

## Sending an SNMP Trap Notification

You can instruct the data collection agent to send an SNMP trap notification when the filter's condition evaluates to true. To do so, use an action statement of the following form:

```
TRAPID = specific_trap_number
```

The data collection agent sends the numeric value entered for *specific\_trap\_number* in the specific trap ID field of the SNMP trap notification packet.

For example:

```
DEFINE_FILTER "AppTrap"  
if ((SUBSYSTEM == "APP") && (MSGID = 11))  
{  
    TRAPID = 5001  
}
```

This filter causes the data collection agent to generate an enterprise-specific SNMP trap with a specific trap ID of 5001 when the agent receives a message with a message ID of 11 from subsystem APP.

The valid keywords in the action part of a filter are shown in Table 6-6.

**Table 6-6 Keywords for Use in Defining an Action**

Construct	Data Type	Description
REMOTE	String	YES or NO. If you specify NO, the message is dropped.
LOCAL	String	<i>/path/filename</i>
COMMAND	String	<i>/path/command arg1 arg2</i>
TRAPID	Integer	Enterprise-specific SNMP trap ID.

## Example

You have one central host and six managed nodes. You have five filters (which we can number F1 through F5) that you want to apply this way:

Host	Filter Combination
H1	F1, F2, F3
H2	F1, F2, F3, F4
H3	F1, F2, F3, F5
H4	F1, F2
H5	F1, F2, F3
H6	F1, F2, F3

- ◆ You want filters F1, F2, and F3 to apply to all hosts, except host H4, so you specify these three in an `LC_GLOBAL` entry.
- ◆ You want host H2 to have access to filter F4. Since F4 is not globally specified, you need a `MANAGED_NODE` entry within which you specify filter F4. Host H2 already knows about filters F1, F2, and F3.
- ◆ You use the same technique for host H3, so that it can use filter F5.

- ◆ You want host H4 to use only filters F1 and F2; that is, you specifically do not want host H4 to inherit the three global filters. To accomplish this, you need a `MANAGED_NODE` entry, within which you “turn off” that inheritance with the `GLOBAL_FILTER` keyword. Doing so requires again individually specifying filters F1 and F2.

**Note:** When you are assigning filters to a specific machine, be aware of the path syntax for the machine. For example, trying to use UNIX path syntax on a Windows NT machine could generate errors.

The following configuration file accomplishes this:

```
DEFINE_FILTER "F1"
if ((USER == "test") || (PID == 12))
{
    REMOTE="NO"
}
DEFINE_FILTER "F2"
if ( LOG_LEVEL == "D" )
{
    LOCAL="/usr/mylogs/locallog"
    REMOTE="NO"
}
DEFINE_FILTER "F3"
if ( MSGBODY >= "network" )
{
    COMMAND="/usr/mybin/notify_admin"
}
node
DEFINE_FILTER "F4"
if ( SUBSYSTEM == "KERNEL" && MSGBODY >= "fatal" )
{
    COMMAND="/usr/mybin/notify_admin"
    TRAPID=123
}
DEFINE_FILTER "F5"
if ((SUBSYSTEM == "NDB") && (MSGID == 12))
{
    REMOTE="NO"
    TRAPID=343
}
LC_GLOBAL
{
    CENTRAL_HOST      = "quahog"
    LOGPREFIX         = "/usr/lmlog"
    FILTER            = "F1"
    FILTER            = "F2"
```

```
        FILTER          = "F3"
    }

    MANAGED_NODE
    {
        HOSTNAME        = "H2"

        FILTER          = "F4"
    }

    MANAGED_NODE
    {
        HOSTNAME        = "H3"

        FILTER          = "F5"
    }

    MANAGED_NODE
    {
        HOSTNAME        = "H4"

        # Do not use global filters
        GLOBAL_FILTER= "NO"
        FILTER          = "F1"
        FILTER          = "F2"
    }
```

## Filter Recommendations

Use enough parentheses in the “if” statement in a filter definition to preclude ambiguities in the evaluation, since precedence rules are not followed strictly during evaluation.

Enclose any string value in double quotes.

Try to use filters as little as possible. Since each log message has to go through all of the filters, throughput can be adversely affected. The `COMMAND` action in particular is very costly.



# 7 Integrating Log Central with an SNMP Manager

Log Central supports Simple Network Management Protocol (SNMP), the de facto industry standard for management of system and network resources. This capability enables monitoring and fault management of log resources or the Log Central system itself from any SNMP-compliant system manager.

Log Central includes an agent that can be used to monitor the Log Central processes on the machine on which it is running. Using this agent, you can instruct an SNMP manager, or an intelligent agent like the BEA Manager Agent Integrator, to poll to periodically check conditions—for example, to ensure that Log Central processes have not died. Log Central data collection agents as well as the Central Collector can also be configured to send event notifications—called *traps* in SNMP terminology—to SNMP managers when critical log messages occur.

This chapter describes how to integrate Log Central with an SNMP manager and includes the following:

- ◆ Facilities for Managing Log Central
- ◆ Setting Up SNMP Management of Log Central
- ◆ Integrating Events Generated by Agent Integrator Polling

# Facilities for Managing Log Central

The facilities provided by Log Central for SNMP systems management are the following:

- ◆ Management Information Base Support for Log Central
- ◆ Process Monitor Agent
- ◆ Central Collector SNMP Trap Generation
- ◆ Data Collection Agent SNMP Trap Generation

## Management Information Base Support for Log Central

A Management Information Base (MIB) describes the attributes of the 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 be formatted in conformity with the SNMP standards. Two MIB files are provided with Log Central that contain information for managing Log Central. These files fully conform to the SNMP standard and are ready for loading into your SNMP manager. The MIB files are:

`bea.asn1`

This file contains MIB definitions for other BEA Manager products as well as Log Central. The Log Central-specific part of this MIB is the Process Monitor MIB, which enables a manager to communicate with the Log Central Process Monitor agent (`pm_snmpd`). The Process Monitor MIB is described in Appendix D, “MIB Reference.”

`bea.asn1`

This file contains MIB definitions for other BEA Manager products as well as Log Central. The Log Central-specific part of this MIB is the Process Monitor MIB, which enables a manager to communicate with the Log Central Process Monitor agent (`pm_snmpd`) Appendix D, “MIB Reference.”

## Process Monitor Agent

The Process Monitor agent (`pm_snmpd`) enables you to monitor the Log Central processes on the machine on which the agent is running. For example, you can find out the number of times the Process Monitor has restarted a process. The attributes of Log Central processes that can be managed with this agent are documented in Appendix D, “MIB Reference.”

The Process Monitor agent can be run as a standalone SNMP agent or as an SNMP Multiplex Protocol (SMUX) subagent under a master agent such as the Agent Integrator.

## Central Collector SNMP Trap Generation

The Central Collector has the ability to generate SNMP trap notifications in response to incoming log messages. This is simple to configure, using the Log Central Console Basic Trap Configuration window, but it is not as fine-grained as the trap generation capability of the data collection agents. The Central Collector can be configured to send a trap based on either the severity of the log message (a measure of the impact on users of the condition that the message is reporting) or, on the message type definition. If you base trap generation on the message definition, a trap is generated if the Trap Enabled attribute is set to `YES`.

## Data Collection Agent SNMP Trap Generation

The data collection agents that you install on the managed nodes have the ability to generate SNMP trap notifications in response to log messages. To activate this feature, you need to define one or more agent filters in the Log Central configuration file (`messaging.conf`). You can define global filters that are used in common by all the log agents, or you can define filters that differ from agent to agent. This is described in Chapter 6, “Host and Filter Configuration.”

# Setting Up SNMP Management of Log Central

To set up Log Central for monitoring by an SNMP management system, do the following:

1. Decide what management information you want to collect.

For example, you might want to consider which log messages that enter the Log Central system should generate SNMP traps for forwarding to your SNMP manager. Also, you might want to consult Chapter D, “MIB Reference,” which provides information on the aspects of the Log Central processes that can be monitored using the Process Monitor agent.

2. Define the destination for SNMP traps in the BEA Manager configuration file.

The TRAP\_HOST entry in the BEA Manager configuration file (`beamgr.conf`) defines the machine and port used by Log Central components as the destination for trap notifications. By default this is the local host. Be sure to edit this file to point to the proper destination for traps. For more information, refer to the “Configuration Files” chapter in the *Agent Integrator Reference Manual*.

3. Configure basic trap generation (if desired).

The Log Central Console has a tool called Basic Trap Configuration that allows you to set up the Central Collector to issue traps under conditions that you specify. This tool is described in Chapter 9, “Using the Log Central Console.”

4. Load the Log Central MIBs into the management system.

You may want to look at the *Agent Connection for M3 and TUXEDO Systems Reference Manual* for examples of how this is done, or else consult the documentation for your management system. The necessary files are:

```
bea.asn1
bea_lc_trap.asn1
```

5. Configure the management system to take appropriate action in response to Log Central trap notifications.

You may want to change the way in which Log Central SNMP traps are displayed on your management console, or the actions that the management

system takes in response to specified events. You might choose to ignore some routine informational notifications. For example, the OpenView Event Configuration window allows you to modify the event configuration to ignore an event, to generate a pop-up notification, or to run a program or script when the event is received. If you are using OpenView, you might also want to create a separate category for Log Central events. For more information, refer to the *Agent Connection for M3 and TUXEDO Systems Reference Manual*.

6. The management system needs to recognize the enterprise of the Log Central traps.

For example, in HP OpenView, you must create a new enterprise entry using the object identifier (OID) for Log Central. The enterprise name is `beaSystemDescr` and the OID is `.1.3.6.1.4.1.140.1.1`.

7. Define Agent Integrator polling rules to poll the Log Central Process Monitor (if desired).

A polling rule is defined by a `RULE_ACTION` entry in the BEA Manager configuration file (`beamgr.conf`). This is described in the *Agent Integrator Reference Manual*. When a user-defined threshold is crossed, the Agent Integrator sends an enterprise-specific trap to the destination specified in the `TRAP_HOST` entry in the BEA Manager configuration file. The enterprise name and OID of the trap are the same as for Log Central traps (see Step 6). If you use this option, you will need to also follow the procedure outlined under “Integrating Events Generated by Agent Integrator Polling.”

8. Configure the Data Collection Agents to generate SNMP traps (if desired).

The Message Sender (a component of the data collection agent on a managed node) can generate traps in response to incoming log messages based on filters that you define in the Log Central configuration file (`messaging.conf`). This allows a more fine-grained selection of which log messages prompt generation of SNMP traps than is possible using the Basic Trap Configuration window on the Log Central Console. You can select messages for SNMP trap generation based on any attribute contained in the message header, or based on text in the body of the message. Agent filter configuration also allows you to specify different SNMP trap generation rules for each managed node, if desired. However, messages cannot be selected for trap generation at the Message Sender based on attributes contained in the message definition (such as Severity or Description) because the definition database is available only to the Central Collector. For more information, refer to Chapter 6, “Host and Filter Configuration.”

# Integrating Events Generated by Agent Integrator Polling

Agent Integrator can be used to do polling of resource attributes monitored by the Process Monitor agent, or other managed resources. To integrate the Agent Integrator threshold-checking activity with the management system, do the following:

1. Set up the Agent Integrator polling rules.

A polling rule is defined by a `RULE_ACTION` entry in the BEA Manager configuration file (`beamgr.conf`). This is described in the *Agent Integrator Reference Manual*.

2. The management system will need to recognize the enterprise of the Agent Integrator traps.

For example, in HP OpenView, you must create a new enterprise entry using the object identifier (OID) for Agent Integrator. The enterprise name is `beaSystemDescr` and the OID is `.1.3.6.1.4.1.140.1.1`.

The Agent Integrator generates an enterprise-specific SNMP trap when the condition defined in the rule is satisfied (evaluates to `TRUE`). The traps generated by the Agent Integrator use the same enterprise identifier as traps generated by Log Central.

3. Configure the management system to recognize the events generated by Agent Integrator.

For example, in HP OpenView, you can add a new event type by selecting Event Configuration → Edit → Add Event. In this case you would use the following as the event number:

`.1.3.6.1.4.1.140.1.1.0.specific_trap_number`

4. Configure the management system to respond appropriately to incoming Agent Integrator events.

This involves essentially the same process as described under Step 5 under “Setting Up SNMP Management of Log Central.”

For more information about Agent Integrator, consult the *Agent Integrator Reference Manual*.

# 8 Starting and Stopping Log Central

To use Log Central, you need to start a Central Collector on your central host and start the agents on the managed nodes.

This chapter discusses the following topics:

- ◆ Starting the Log Central Data Collection System
- ◆ Stopping the Log Central Data Collection System
- ◆ Displaying Log Central System Information

The procedures for starting Log Central described in this chapter assume that you have completed the initial configuration steps discussed in Chapter 2, “Getting Started.”

# Starting the Log Central Data Collection System

The procedure for starting Log Central follows:

1. Run the `start_messaging` command on the central host to start the Central Collector.

After starting the Central Collector, the `start_messaging` process continues to run and acts as the server of the host and filter configuration to the `start_messaging` process on managed nodes. The `start_messaging` command also starts the Process Monitor (the `proc_monitor` process). The Process Monitor monitors the `start_messaging` process to ensure that it continues to run and restarts it if it dies. The `start_messaging` process in turn monitors the `proc_monitor` process and restarts it if it dies.

Because Log Central processes on the central host provide Log Central agents with their configuration information when they are started, the Central Collector must be started before any of the data collection agents are started

This is described under “Starting Log Central on a Central Host.”

**Note:** If you are using a backup Central Collector, start it at this time also.

2. On each managed node, use the `start_messaging` command to start the Message Sender and Process Monitor.

This is described under “Starting Log Central on the Managed Nodes.”

3. On each managed node, start a Log Monitor process for each log resource that you wish to manage.

This is described under “Starting Log Monitor.”



## Starting Log Central on a Central Host

The `start_messaging` command starts Log Central components on the central host. The minimum set of components started by the `start_messaging` command on the central host are:

- ◆ Message Receiver (`msg_receiver`)
- ◆ Message Processor (`msg_processor`)
- ◆ Process Monitor (`proc_monitor`)
- ◆ Message sender (`msg_sender`)

To use the `start_messaging` command on the central host, enter the following command:

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

where:

`-f config_file`

Uses *config\_file* as the Log Central configuration file in place of the default (*install\_dir/etc/messaging.conf*).

`-q`

Makes the process “quiet”; that is, no informational messages are displayed.

`-v`

Specifies the verbose option, which displays informational and debug messages.

`-h`

Prints out detailed usage information.

`central_host`

Specifies the host name of the machine acting as the primary central host. If not specified, the current host is considered to be the central host. This parameter must be specified when starting the Log Central system on the backup host or a managed node.

*backup\_central\_host*

Specifies the host name of the machine where the backup Central Collector is running. Data collection agents connect to the backup Central Collector if the Central Host cannot be reached.

**Note:** You need to specify a backup central host with the `start_messaging` command only if you are starting a data collection agent on the central host in addition to a Central Collector.

Because Log Central processes on the central host provide Log Central agents with their configuration information when they are started, the Central Collector must be started before any of the data collection agents are started.

After being invoked, the `start_messaging` process continues to run and can be used to stop the Central Collector (and other central host Log Central processes) by invoking the `stop_messaging` command. Use of the `stop_messaging` command is described in “Stopping the Log Central Data Collection System.”

The `start_messaging` process must be able to find the Log Central configuration file (`messaging.conf`). The default location of this file is:

◆ `install_dir/etc/messaging.conf` on UNIX systems

◆ `install_dir\etc\messaging.conf` on Windows NT systems

The `install_dir` variable is the directory under which you installed Log Central.

If you want `start_messaging` to use a configuration file that does not reside in the default location, invoke `start_messaging` with the `-f` option to provide a path to the file.

When starting a backup Central Collector, you must specify the `central_host` parameter. It is not necessary to specify the `central_host` when starting the primary Central Collector because `start_messaging` assumes that the machine it is invoked on is the central host if this parameter is not specified.

Once the `start_messaging` process is started on the central host, changes made to the Log Central configuration file will not take effect until the next time the Log Central processes are started using the `start_messaging` command.

## Starting Log Central on the Managed Nodes

A managed node is any machine remote from the Central Collector that has log resources that you want to monitor. To start the Log Central processes on a managed node, do the following:

1. Start the Message Sender and Process Monitor by invoking the `start_messaging` command. The syntax for the `start_messaging` command, when used on a managed node, follows:

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

where:

`-q`

Makes the process “quiet”; that is, no informational messages are displayed.

`-v`

Specifies the verbose option, which displays informational and debug messages.

`-h`

Prints out detailed usage information.

`central_host`

Specifies the host name of the machine where the primary Central Collector is running. If not specified, the current host is assumed to be the central host.

`backup_central_host`

Specifies the host name of the machine where the backup Central Collector is running. Data collection agents connect to the backup Central Collector if the primary Central Collector cannot be reached.

**Note:** You must have already invoked the `start_messaging` command on the central host.

2. Start a Log Monitor process for each log resource that you want to monitor.

This is described under “Starting Log Monitor.”

**Note:** You must invoke the `start_messaging` command on the managed node before starting the Log Monitor processes.

To start the Log Central subsystem on a managed node, you must supply the name of the central host to the `start_messaging` process. The `start_messaging` process on the managed node connects to the `start_messaging` process running on the central host, using the `udp` service defined by the environment variable `BEA_LC_CONF_SERVICE`. The `start_messaging` process on the managed node then downloads the local host's configuration. If `BEA_LC_CONF_SERVICE` is not defined, a service called `lc_conf` is used by default. The `start_messaging` process then starts the Log Central subsystem according to the configuration received from the central host.

## Starting Log Monitor

You must start a Log Monitor process for each log that you want to monitor on a managed node. The Log Monitor reads the logs generated by the managed resource, such as a computer system, a BEA TUXEDO application, or a relational database system. Log Monitor maps the attributes in the managed resource log messages to attributes in Log Central messages. Messages are then placed in the Message Sender's queue for forwarding to the Central Collector.

You can instruct Log Monitor to map log messages to Log Central message format for forwarding to the Central Collector in three ways:

- ◆ Using a predefined mapping, which is described under “Starting Log Monitor with Predefined Mappings.”
- ◆ Using a configuration file in which you have defined a mapping, which is described in “Starting Log Monitor with Mappings in a Configuration File.”
- ◆ By specifying the mapping options on the command line, which is described in “Starting Log Monitor with Mapping Specified on the Command Line.”

### Starting Log Monitor with Predefined Mappings

Predefined mappings are available for integrating the following log resources into the Log Central system:

- ◆ TUXEDO—Used for mapping BEA TUXEDO logs
- ◆ NTEVENTLOG—Windows NT event log

- ◆ ORACLE—Oracle alert log
- ◆ LC—Log Central temporary log files

**Note:** For usage of LC, see “Using the LC Predefined Mapping.”

To invoke Log Monitor with a predefined mapping, use the following command:

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

where:

*-i filename*

This option specifies the incoming message file to use. You can also use a dash (*-i -*) to specify standard input.

*-P predefined\_mapping*

This option tells Log Monitor to use a built-in message format mapping. The argument can be any one of the following:

```
TUXEDO
LC
NTEVENTLOG
ORACLE
```

*-t time*

This option specifies the amount of time to wait between forwards. The default is one second (that is, 1). Use 0 to forward once and then stop.

*-p pattern*

This option instructs the program to forward only lines that match *pattern*, which may simply be a string or it may use special characters. These are defined in Chapter 4, “Integrating Logs into Log Central.”

*-x pattern*

This option instructs the program not to forward lines that match *pattern*, which may simply be a string or it may use special characters. These are defined in Chapter 4, “Integrating Logs into Log Central.”

*-e entityname*

This option specifies the entity name to be used by the `log_monitor` process to register to the `proc_monitor` process. The default value is `log_monitor`. All Log Monitors on one managed node must have unique entry names. If the `log_monitor` process is run as a daemon (with *-t 0*), then the entity name option is not used.

The `-p` option can be used to select only certain messages for forwarding. The `-x` option can be used to select messages to be dropped. For more information, refer to Chapter 4, “Integrating Logs into Log Central.”

For details about the predefined mappings, refer to Appendix G, “Predefined Log Mapping.”

### USING THE LC PREDEFINED MAPPING

Log Central processes create temporary log files, which exhibit a common log file format. Normally the contents of these files make their way into the Log Central database through normal operation of the system. 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 host is unavailable. When the Central Collector becomes available again, the Message Sender automatically forwards the log messages from the temporary file to the Central Collector. However, if the Message Sender dies before it can recover the file, the temporary file may not be recovered automatically. If this happens, you can recover the contents of this temporary file by starting a Log Monitor to read the temporary log file, using the LC mapping.
- ◆ The Message Receiver on the central host writes incoming messages to an intermediate file. The Message Processor reads this file to insert these messages into the database. If for some reason an intermediate file does not get processed by the Message Processor, you can recover the contents of the intermediate file by invoking Log Monitor to read the intermediate file, using the LC mapping.

These are the only situations where you would start Log Monitor with the LC mapping.

## Starting Log Monitor with Mappings in a Configuration File

A configuration file is simply a list of Log Monitor filters, each on a separate line. A log message will be forwarded by the Log Monitor if it is selected by at least one of the filters in the configuration file.

For information on constructing mappings in a Log Monitor configuration file, refer to Chapter 4, “Integrating Logs into Log Central.”

The syntax for invoking Log Monitor with a configuration file follows:

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

where:

`-f config_filename`

This option specifies the configuration file from which to read filters.

`-i filename`

This option specifies the incoming message file to use. You can also use a dash (`-i -`) to specify standard input.

`-t time`

This option specifies the amount of time to wait between forwards. The default is one second (that is, 1). Use 0 to forward once and then stop.

`-c`

This option applies the commands in the configuration file only up to the first match in the configuration file.

For examples, see Chapter 4, “Integrating Logs into Log Central.”

`-e entityname`

This option specifies the entity name to be used by the `log_monitor` process to register to the `proc_monitor` process. The default value is `log_monitor`. All Log Monitors on one managed node must have unique entry names. If the `log_monitor` process is run as a daemon (with `-t 0`), then the entity name option is not used.

## Starting Log Monitor with Mapping Specified on the Command Line

The syntax for invoking Log Monitor by specifying all options on the command line follows:

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

where:

`-i filename`

This option specifies the incoming message file to use. You can also use a dash (`-i -`) to specify standard input.

`-t time`

This option specifies the amount of time to wait between forwards. The default is one second (that is, 1). Use 0 to forward once and then stop.

- M *log\_level*  
This option specifies the logging level to use in the message. Logging level is a single character that must be one of the following:
  - N—A normal message
  - V—A verbose message
  - D—A debug message
  - S—A special message
- m *subsystem*  
This option specifies the subsystem name to use in the message. The default is none.
- d *msgid*  
This option specifies the message ID to use in the message. The default is 1000.
- n *function*  
This option specifies the function name to use in the message. The default is none.
- u *userID*  
This option specifies the user ID to use in the message. The default is the current user (that is, the owner of the `log_monitor` process).
- o *hostname*  
This option specifies the host name to use in the message. The default is the machine on which Log Monitor is running.
- I *processID*  
This option specifies the process ID to use in the message. The default is the process ID of the daemon process invoked by the Log Monitor command.
- b *body*  
This option specifies the body of the message.
- D *date %f"format"*  
This option specifies the date to use in the message. For information on specifying the date format, see Chapter 4, “Integrating Logs into Log Central.”



-p *pattern*

This option instructs the program to forward only lines that match *pattern*, which may simply be a string or it may use special characters. These are defined in Chapter 4, “Integrating Logs into Log Central.”

-x *pattern*

This option instructs the program not to forward lines that match *pattern*, which may simply be a string or it may use special characters. These are defined in Chapter 4, “Integrating Logs into Log Central.”

-T *transactionID*

This option specifies the transaction ID to use in the message. The default is 0.

-P *predefined\_mapping*

This option tells Log Monitor to use a built-in message format mapping. The argument can be any one of the following:

TUXEDO  
LC  
NTEVENTLOG  
ORACLE

-e *entityname*

This option specifies the entity name to be used by the `log_monitor` process to register to the `proc_monitor` process. The default value is `log_monitor`. All Log Monitors on one managed node must have unique entry names. If the `log_monitor` process is run as a daemon (with `-t 0`), then the entity name option is not used.

-S

This option specifies one or more separators to be used to calculate field values in the input message file for corresponding `%F` specifiers on a command line. (The `%F` format symbol is described in Chapter 4, “Integrating Logs into Log Central.”)

If more than one separator is specified, all are used to count the fields. If a message starts with a separator, the text between the first and the second separator is counted as field number 1; fields are numbered starting with 1 (not 0). If a message does not start with a separator, the first field consists of the text up to the first separator.

For examples, see Chapter 4, “Integrating Logs into Log Central.”

A number of these options refer to specific attributes in a Log Central log message, such as message ID or process ID. The Log Central message format is described in Appendix A, “Message Format.”

For information on how to use these options to construct mappings, consult Chapter 4, “Integrating Logs into Log Central.”

## Stopping the Log Central Data Collection System

To stop the Log Central data collection system, you must issue the `stop_messaging` command on each node where Log Central components are running.

The syntax of the `stop_messaging` command follows:

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

where:

- q                      Makes the process “quiet”; that is, no informational messages are displayed.
- v                      Specifies the verbose option, which displays informational and debug messages.
- h                      Prints out detailed usage information.

When Log Monitor starts, it registers with the `proc_monitor` process. This enables the `stop_messaging` process to stop the Log Monitor processes when you issue the `stop_messaging` command to shut down the Log Central system. No separate command is required to shut down the Log Monitor processes.

# Displaying Log Central System Information

Log Central includes a command, `show_config`, that enables you to display the current state of your Log Central configuration file, compile the configuration file, or display Log Central shared memory information.

Depending upon the arguments or options you enter, the `show_config` command compiles the configuration file or dumps the Log Central shared memory information to `stdout`. You can use this command to see if all processes are up or what processes are no longer running and to check for other system maintenance indicators.

The syntax of the `show_config` command follows:

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

where:

`-c`

Check syntax errors in configuration file.

`-f config_filename`

This option specifies the name of the Log Central configuration file to use with the `-c` option. If this option is not specified, `install_dir/etc/messaging.conf` is taken as the default. The `install_dir` variable is the directory in which Log Central was installed.

`-g`

Dumps the Log Central global information from shared memory to `stdout`.

`-p`

Dumps the `proc_monitor` shared memory information about the processes being monitored to `stdout`.

`-d`

Dumps the whole Log Central shared memory to `stdout`. This option is equivalent to using both the `-g` and `-p` options.

`-e entityname`

This option specifies the entity name to be used by the `log_monitor` process to register to the `proc_monitor` process. The default value is `log_monitor`. All Log Monitors on one managed node must have unique entry names. If the `log_monitor` process is run as a daemon (with `-t 0`), then the entity name option is not used.

`-h`

Prints out detailed usage information.

### Example

Before using your configuration file at run time, you may wish to check for syntax errors. To validate your configuration file, run the following command:

```
show_config -c -f config_file
```

In the preceding command, if the `-f` option is not specified, the file `install_dir/etc/messaging.conf` is used by default.

The `show_config` command displays syntax errors to `stdout`, showing line and character position.

**Note:** Avoid the use of tabs in your configuration file because they can cause the character positions reported to be inaccurate.

# 9 Using the Log Central Console

The Log Central Console is a graphical user interface that provides your main point of access for controlling the flow of information in Log Central, monitoring incoming log messages, and carrying out other management tasks. The Console consists of a set of tools that you can access from the Log Central Launch Panel.

This chapter covers the following topics:

- ◆ Invoking the Launch Panel
- ◆ Using the Message Browser
- ◆ Using the Message Definition Editor
- ◆ Using the Basic Trap Configuration Tool
- ◆ Using the Storage Maintenance Tool

## Invoking the Launch Panel

To invoke the Log Central Console, type `lc_launch` at a command prompt on the central host.

When you start the Console, the Launch Panel appears first. The Launch Panel provides access to the Console tools of Log Central.



You can also invoke the Console with a number of options. You may want to specify any of the following:

- ◆ Message processor port number
- ◆ The central host, if you are invoking the launch panel from other than the central host
- ◆ Some maximum number of messages to display
- ◆ The font and font size to use within all the windows of the Console
- ◆ The Web browser to use for displaying help about Log Central

To change any of the `lc_launch` defaults, use the following syntax:

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

where:

*port*

Specifies the message processor's port number. The default is 7001.

*central\_host*

Specifies the host name of the machine where the Central Collector is running. If not specified, the current host is considered as the central host.

*msgs*

Specifies the maximum number of messages to display in the Message Browser. The default is 1000.

*fontname*

Specifies the font to use wherever text appears in the Console. The default is Times Roman.

*fontsize*

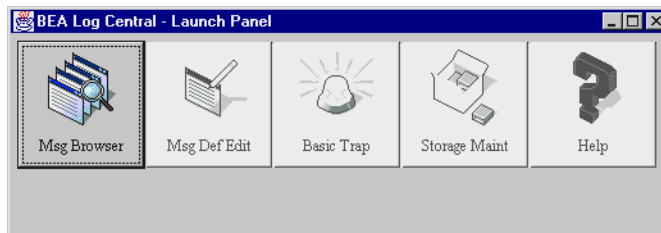
Specifies the font size to use wherever text appears in the Log Central Console. The default is 12 points.

*browser*

Specifies the Web browser to use for displaying help about Log Central. Use either the name of an executable within the current path specification, or specify an executable together with its complete path name. The default is Netscape.

## Using the Message Browser

The Message Browser displays Log Central messages that are generated by the resources you are monitoring. Invoke the Message Browser main window by selecting the `Msg Browser` button on the Launch Panel.



The subjects discussed in this section are as follows:

- ◆ Message Browser Main Window
- ◆ How to Monitor Incoming Messages
- ◆ How to Perform Historical Queries of the Message Database
- ◆ How to Delete Messages
- ◆ How to Acknowledge Messages
- ◆ How to Remove Acknowledgment from Messages
- ◆ How to Filter Messages

- ◆ How to Change the Message Layout in the Main Window
- ◆ How to Change Message Colors
- ◆ How to View Message Details
- ◆ How to Generate Reports

## Message Browser Main Window

Figure 9-1 shows the basic look and presentation of the Message Browser main window.

**Figure 9-1 Message Browser Main Window**

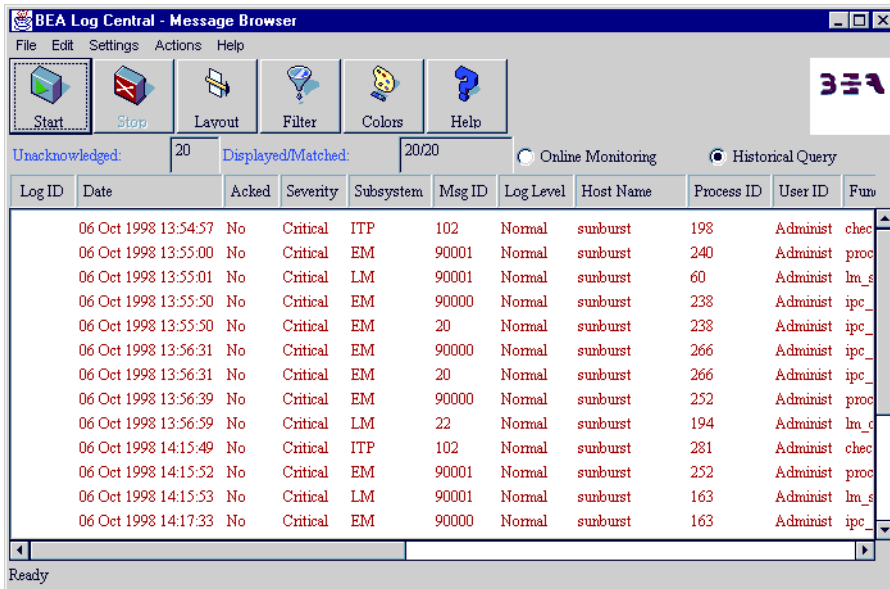










Table 9-1 describes the fields of this window.

**Table 9-1 Message Browser Main Window Fields**

Field	Description
Menus	The File, Edit, Settings, Actions, and Help menus provide access to Message Browser tools.
Toolbar	The toolbar buttons provides access to frequently used actions. Refer to Table 9-2 for a description of the toolbar buttons.
Unacknowledged	This field displays the number of messages in the main window scrollable display that have not been acknowledged. A typical use for the acknowledgment attribute is to chart which system problems are currently being resolved or actively investigated.
Displayed/Matched	When you execute a search of the database for messages matching a search filter, the main window displays the number of messages that match your search criteria and the number of messages displayed. These numbers differ if the search found more messages than the maximum that can be displayed. The maximum number of messages that can be displayed by the Message Browser is configurable, as detailed under “Invoking the Launch Panel.”
Online Monitoring	Select Online Monitoring to monitor log messages “in real time” as they arrive at the Central Collector, with the most recent message at the top of the window.
Historical Query	Select Historical Query to search the Log Central database for log messages that match criteria that you define.
Message pane	Displays messages that meet the filtering criteria.

Table 9-2 describes the Message Browser toolbar buttons available for frequently used actions.

**Table 9-2 Message Browser Toolbar Buttons**

Tool Button or Field	Action
	Applies the settings defined in the Filter Settings window to select messages to be displayed, either incoming messages as they arrive at the Central Collector, or a query of messages in the Log Central database.
	Stops retrieval of messages. This button is grayed out if you have not yet started message retrieval.
	Invokes the Layout Definition window. You can use this window to specify which fields in messages to display and which order to place the columns in the table.
	Invokes the Filter Settings window. You can use this window to: <ul style="list-style-type: none"> <li>◆ Switch between online monitoring and historical queries.</li> <li>◆ Define the criteria that are used to determine which messages are displayed in the main window.</li> <li>◆ Select the sort order for message display in a historical query.</li> </ul>
	Invokes the Message Colors window. You use this window to change the mapping of colors to message severity level.
	Invokes the help page.

**Note:** All options other than Stop and Help are grayed out if the Browser is either actively receiving incoming messages or is carrying out a database query. Select Stop to access the other options.

## How to Monitor Incoming Messages

You can monitor log messages “in real time” as they arrive at the Central Collector. To monitor incoming messages, perform the following steps:

1. Select the Online Monitoring radio button on the Message Browser main window.
2. Select Start from the toolbar on the Message Browser main window.

Alternatively, you can select the Online Monitoring radio button on the Filter Settings window, and then select Start from the toolbar when you return to the Message Browser main window.

Messages are retrieved that satisfy the filter settings defined in the Filter Settings window.

The scrollable display in the main window accumulates incoming messages until it contains its maximum number of messages. At that point, the oldest messages are dropped from the display as new messages arrive. New messages appear at the top of the tabular list.

To filter incoming messages, see “How to Filter Messages.”

## How to Perform Historical Queries of the Message Database

You can search the Log Central database for log messages that match criteria that you define. To do a historical query of the message database, perform the following steps:

1. Select the Historical Query radio button on the Message Browser main window.
2. Select Start from the toolbar on the Message Browser main window.

To filter historical queries, see “How to Filter Messages.”

Alternatively, you can select the Historical Query radio button on the Filter Settings window, and then select Start from the toolbar when you return to the Message Browser main window.

Messages are retrieved that satisfy the filter settings defined in the Filter Settings window.

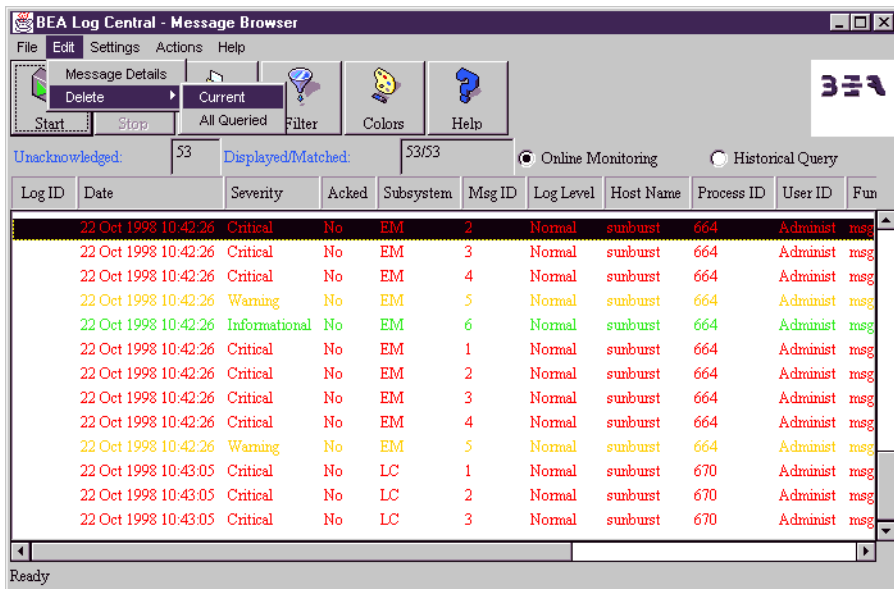
The default is for retrieved messages to appear on your screen in chronological order, with the most recent at the top of the screen, from the last query made. You can change this default with filtering, as described in “How to Filter Messages.”

## How to Delete Messages

From the Message Browser main window, you can delete a log message as follows:

1. If you are actively monitoring online messages, select Stop.
2. Select the message that you want to delete.
3. Select Edit → Delete from the Message Browser main window, as shown in Figure 9-2.

**Figure 9-2** Selecting Delete from the Message Browser Main Window



A submenu allows you to delete either the current message or all messages that were returned in the last query.

## How to Acknowledge Messages

When messages arrive at the Central Collector, they are by default not acknowledged. You can change the acknowledged value to “Yes” from the Message Browser. A typical use for this attribute is to chart which system problems are currently being resolved or actively investigated.

From the Message Browser main window, you can acknowledge a log message as follows:

1. If you are actively monitoring online messages, select Stop.
2. Select the message that you want to acknowledge.
3. Select Actions → Acknowledge.

**Figure 9-3** Selecting Actions → Acknowledge from Message Browser Main Window



A submenu allows you to change the acknowledgment of either the current message or all main window messages retrieved in the last query.

Once the message has been acknowledged, “Yes” appears in the Acknowledged column.

## How to Remove Acknowledgment from Messages

From the Message Browser main window, you can change the acknowledgment of a log message from “Yes” to “No” as follows:

1. Select the message whose acknowledgment you want to remove.
2. Select Actions → Clear → Acknowledge.

A submenu allows you to change the acknowledgment of either the current message or all main window messages retrieved in the last query.

Once the acknowledgment of the message has been removed, “No” appears in the Acknowledged column.

## How to Filter Messages

You can use filters to specify selection criteria to:

- ◆ Define which incoming messages are selected for display when doing online monitoring
- ◆ Define which messages to retrieve in historical database queries

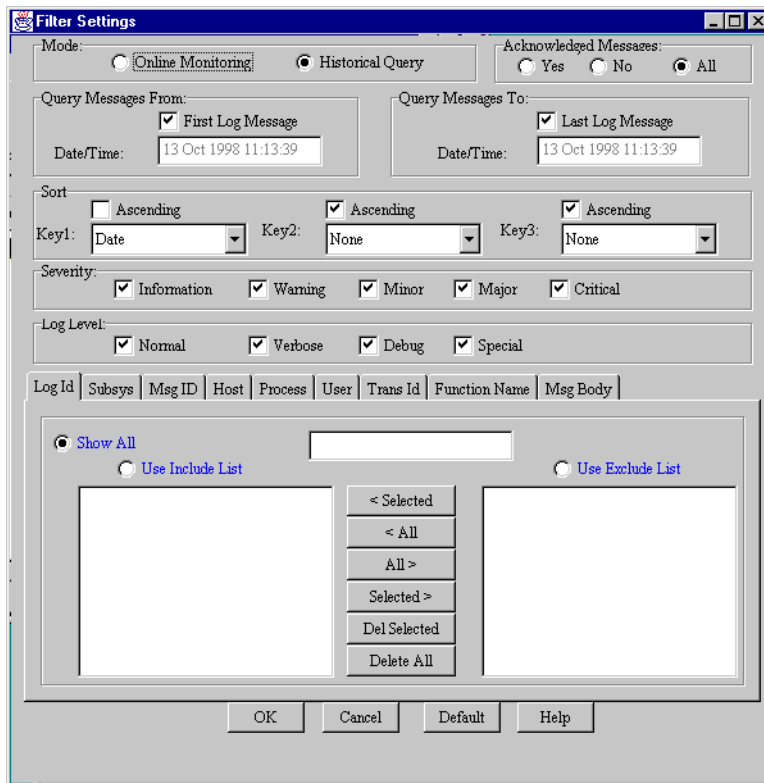
From the Message Browser main window, you can filter log messages as follows:

1. Select the Filters button on the main window.

Alternatively, you can select the Settings → Browser Filter menu option.

The Filter Settings window appears on the screen, as shown in Figure 9-4.

**Figure 9-4 Filter Settings Window**



From the Filters Settings window, you can switch between online message monitoring and historical queries of the Log Central database.

2. Define the selection criteria from which to view messages in the message browser.

To filter incoming messages, select Online Monitoring. To filter messages already stored in the Log Central database, select Historical Query.

If you select Online Monitoring, the date and sorting sections of the form are grayed out, as shown in Figure 9-5.

**Figure 9-5 Filter Settings Window with Portions Grayed Out**

The screenshot shows the 'Filter Settings' dialog box. At the top, the 'Mode' section has two radio buttons: 'Online Monitoring' (selected) and 'Historical Query'. To the right, the 'Acknowledged Messages' section has three radio buttons: 'Yes', 'No', and 'All' (selected). Below this, the 'Query Messages From' and 'Query Messages To' sections are grayed out. Each has a 'First Log Message' or 'Last Log Message' checkbox (checked) and a 'Date/Time' field showing '13 Oct 1998 11:13:39'. The 'Sort' section is also grayed out, showing three 'Ascending' checkboxes and three dropdown menus for 'Key1', 'Key2', and 'Key3'. The 'Severity' section has five checkboxes: 'Information', 'Warning', 'Minor', 'Major', and 'Critical', all of which are checked. The 'Log Level' section has four checkboxes: 'Normal', 'Verbose', 'Debug', and 'Special', all of which are checked. At the bottom, there is a tabbed interface with tabs for 'Log Id', 'Subsys', 'Msg ID', 'Host', 'Process', 'User', 'Trans Id', 'Function Name', and 'Msg Body'. Below the tabs, there are three radio buttons: 'Show All' (selected), 'Use Include List', and 'Use Exclude List'. To the right of these are two empty list boxes and a set of buttons: '< Selected', '< All', 'All >', 'Selected >', 'Del Selected', and 'Delete All'. At the very bottom are four buttons: 'OK', 'Cancel', 'Default', and 'Help'.

Select the sort order for messages retrieved in historical database queries. You can nest up to three sorts, mixing ascending and descending sort order.

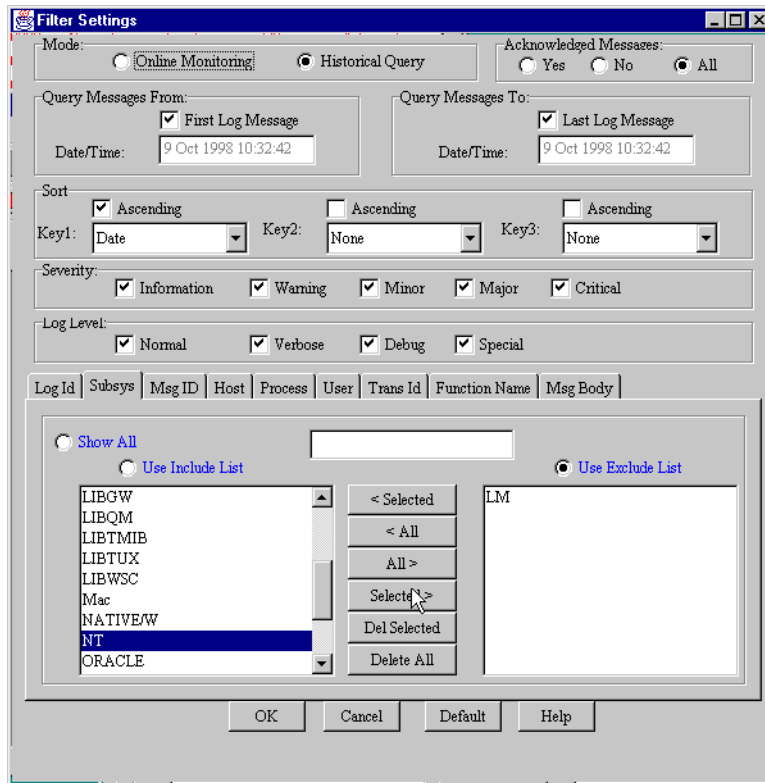
3. If you want to specify different filtering criteria for any of the nine fields shown in the tabs across the middle of the Filter Settings window, select that tab.

For example, the Subsys tab displays a list of the subsystem entries in the Log Central database, and you can use the buttons between the two windows (Include List and Exclude List) to move selections from one window to the other, or delete selections.



Figure 9-6 shows the LM subsystem in the exclude list and NT selected and ready to be moved to the exclude list.

**Figure 9-6 Filter Settings Window, Include and Exclude Selections**



For other tabs, enter your specification in the field above the two list boxes, and use the appropriate button between the list windows to move this specification to the list you want. For example, Figure 9-7 shows the Msg ID tab selected and message 90000 entered.

**Figure 9-7 Filter Settings Window, Include Selection**

**Filter Settings**

Mode: ☒ Online Monitoring ☐ Historical Query

Acknowledged Messages: ☐ Yes ☐ No ☒ All

Query Messages From: ☒ First Log Message  
Date/Time: 3 Nov 1998 9:57:41

Query Messages To: ☒ Last Log Message  
Date/Time: 3 Nov 1998 9:57:41

Sort: ☐ Ascending ☐ Ascending ☐ Ascending  
Key1: Date Key2: None Key3: None

Severity: ☒ Information ☒ Warning ☒ Minor ☒ Major ☒ Critical

Log Level: ☒ Normal ☒ Verbose ☒ Debug ☒ Special

Log Id Subsys Msg ID Host Process User Trans Id Function Name Msg Body

☐ Show All ☒ Use Include List ☐ Use Exclude List

9000

< Selected  
< All  
All >  
Selected >  
Del Selected  
Delete All

OK Cancel Default Help

Figure 9-8 shows this message moved to the include list.

**Figure 9-8 Filter Settings Window, Include Selection Made**

The screenshot shows the 'Filter Settings' window with the following configuration:

- Mode:** ☒ Online Monitoring, ☐ Historical Query
- Acknowledged Messages:** ☐ Yes, ☐ No, ☒ All
- Query Messages From:** ☒ First Log Message, Date/Time: 3 Nov 1998 9:57:41
- Query Messages To:** ☒ Last Log Message, Date/Time: 3 Nov 1998 9:57:41
- Sort:** Key1: ☐ Ascending, Date; Key2: ☐ Ascending, None; Key3: ☐ Ascending, None
- Severity:** ☒ Information, ☒ Warning, ☒ Minor, ☒ Major, ☒ Critical
- Log Level:** ☒ Normal, ☒ Verbose, ☒ Debug, ☒ Special
- Log Id | Subsys | Msg ID | Host | Process | User | Trans Id | Function Name | Msg Body**
- Show All:** ☐ Show All, ☒ Use Include List, ☐ Use Exclude List
- Include List:** 9000, 9001
- Exclude List:** (Empty)
- Buttons:** < Selected, < All, All >, Selected >, Del Selected, Delete All
- Bottom Buttons:** OK, Cancel, Default, Help

4. Select either Use Include List or Use Exclude List.

For Use Include List, the specified features are used to determine which messages to display. For Use Exclude List, those messages that meet the selected criteria are specifically excluded from the display.

5. Repeat steps 3 to 4 for as many tabs as you want.
6. When you have made as many specifications as you want, select OK.

## How to Change the Message Layout in the Main Window

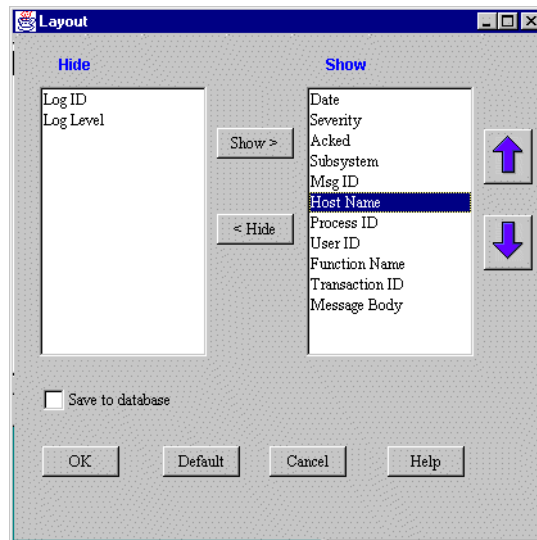
From the Message Browser main window, you can change the layout to define which log attributes are displayed in the main window or to change the order of the attributes in the tabular display:

1. Select Layout from the Message Browser main window.

Alternatively, you can select the Settings → Layout Definition menu option.

The Layout window appears on the screen.

**Figure 9-9 Message Browser Layout Window**



2. Select the components that you want to show or hide.

You can move components into the appropriate column by selecting the Show or Hide arrows on the main Layout window. You can move components up and down with the blue arrows to the right. The order you specify in the Show panel

determines the order these columns appear in the message panel of the Message Browser main window, shown in Figure 9-1.

3. Select OK.

**Note:** If you want these selections as your default settings, select Save to Database before selecting OK.

## How to Change Message Colors

By default, messages appear in the Message Browser main window in different colors, depending on severity. For example, you might want messages of critical severity to appear in red, warnings in yellow, and so on. The default color-mapping is as follows.

Severity	Color
Informational	Green
Warning	Yellow
Minor	Blue
Major	Magenta
Critical	Red

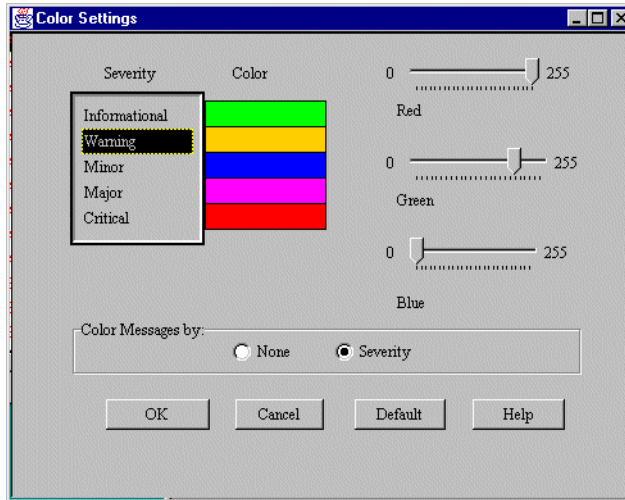
To change the mapping of colors to indicate message severity level:

1. Select Colors from the Message Browser main window.

Alternatively, you can select the Settings → Message Colors menu option.

The Color Settings window appears on the screen.

**Figure 9-10 Message Browser Color Settings Window**



2. In the Severity column, select the severity level whose color you want to change.
3. Move the RGB sliders to get the color you want.  
As you move the sliders, the color in the Color column changes.
4. Repeat steps 2 and 3 for as many severity levels as you want to change.  
**Note:** If you do not want to map colors to severity levels, select the None radio button.
5. Select OK.

## How to View Message Details

Message definitions are stored in the Log Central database. These message definitions are described in Chapter 5, “Creating and Loading Message Definitions.” You can access this message detail information for a particular message, such as a description of the event that triggered the message and recommended action to resolve a problem condition.

1. Select the message that you want to access information about.
2. Select Edit → Message Details from the Message Browser main window.

The Message Details window appears showing the message details stored in the Message Definition database.

## How to Generate Reports

You can generate two types of reports from the Message Browser:

- ◆ Summary reports
- ◆ Detail reports

For both, the Message Browser uses the current filter criteria, as defined in the Filter Settings window, to select messages on which to generate a report. The generated report is an HTML file created in *install\_dir*\web\applet (*install\_dir*/web/applet on UNIX), where *install\_dir* is the directory in which you installed Log Central. Viewing this file requires a Web browser. Make sure that you have made a Web browser accessible to the Console. For information on how to do this, refer to “Invoking the Launch Panel.”

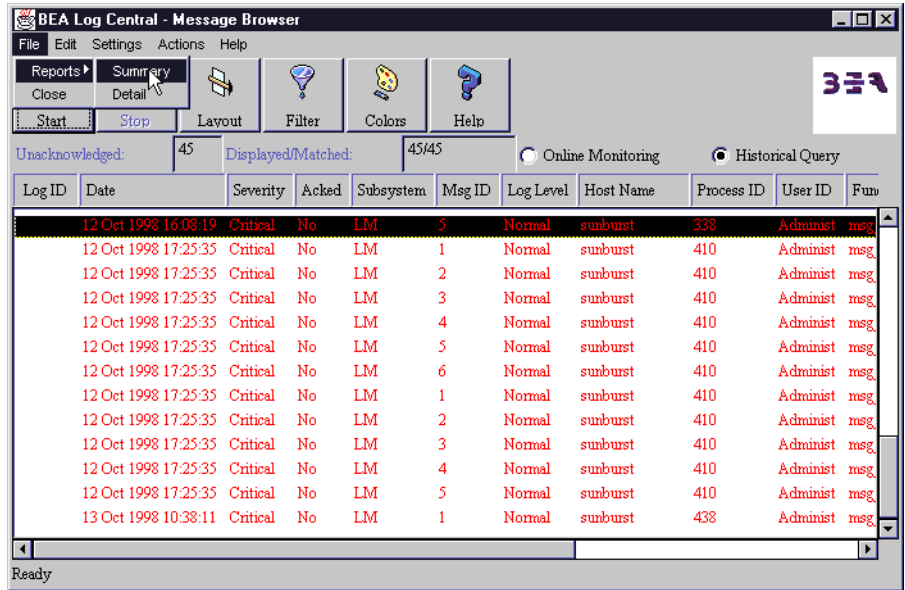
## How to View Summary Reports

To view summary reports in the Message Browser, perform the following steps:

1. Specify the filtering criteria, as described in “How to Filter Messages.”
2. Start your query, as described in “How to Perform Historical Queries of the Message Database.”
3. Select File → Reports → Summary.

Figure 9-11 shows a message selected, and the summary reports option chosen.

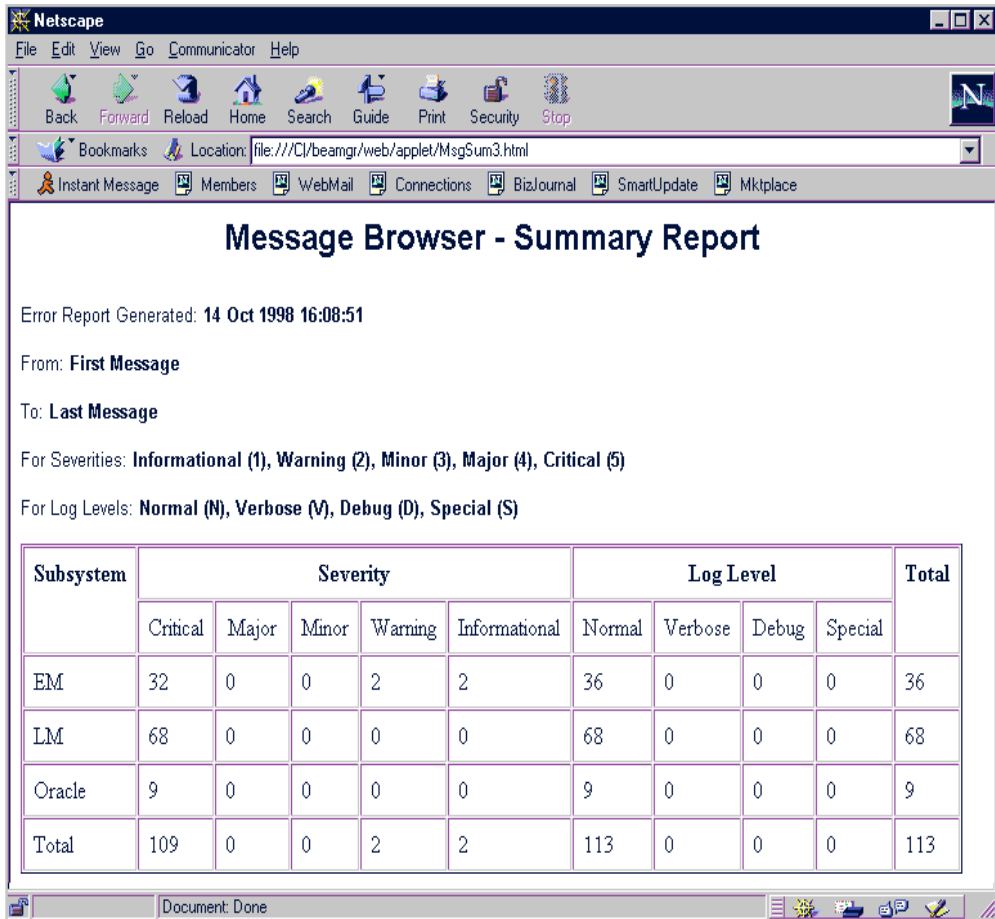
**Figure 9-11 Message Browser Window, Choosing Summary Report**





A Web browser window appears, displaying the requested report.

**Figure 9-12 Message Browser Summary Report**

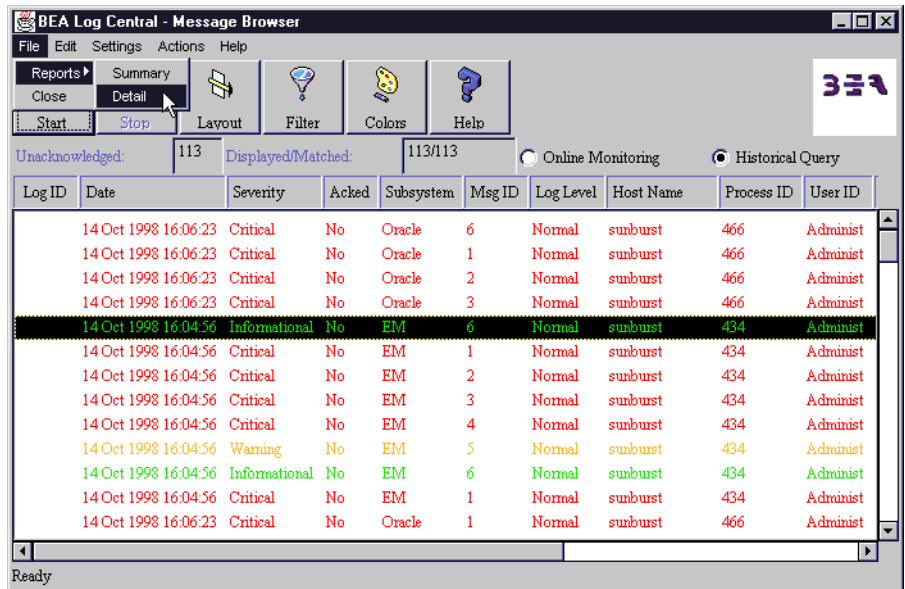


## How to View Detail Reports

To view detail reports in the Message Browser, perform the following steps:

1. Specify the filtering criteria, as described in “How to Filter Messages.”
2. Start your query, as described in “How to Perform Historical Queries of the Message Database.”
3. Select File → Reports → Details

**Figure 9-13 Message Browser Window, Choosing Detail Report**



A Web browser window appears, displaying the requested report.

**Figure 9-14 Message Browser Detail Report**

**Message Browser - Detail Report**

Error Report Generated: **14 Oct 1998 16:22:07**

From: **First Message**

To: **Last Message**

For Severities: **Informational (1), Warning (2), Minor (3), Major (4), Critical (5)**

For Log Levels: **Normal (N), Verbose (V), Debug (D), Special (S)**

SEV	DATE	SUBSYS	MSG ID	MESSAGE SUMMARY	PID	USER ID	LOG LVL	HOST
Informational	1998-07-15 08:48:25.0	CMDTUX	4350	BBL started on machine - Release val	20296	sachin	Normal	jellyfish
CMDTUX_CAT:4350: INFO: BBL started on SITE2 - Release 6400								
Informational	1998-07-15 08:48:14.0	LIBTUX	262	Standard main starting	20298	sachin	Normal	jellyfish
LIBTUX_CAT:262: INFO: Standard main starting								
Informational	1998-07-15 08:48:14.0	LIBTUX	262	Standard main starting	20298	sachin	Normal	jellyfish
LIBTUX_CAT:262: INFO: Standard main starting								
Informational	1998-07-15 08:48:25.0	CMDTUX	4350	BBL started on machine - Release val	20296	sachin	Normal	jellyfish
CMDTUX_CAT:4350: INFO: BBL started on SITE2 - Release 6400								
Informational	1998-07-15 08:48:14.0	LIBTUX	262	Standard main starting	20298	sachin	Normal	jellyfish
LIBTUX_CAT:262: INFO: Standard main starting								
Informational	1998-07-15 08:48:25.0	CMDTUX	4350	BBL started on machine - Release val	20296	sachin	Normal	jellyfish

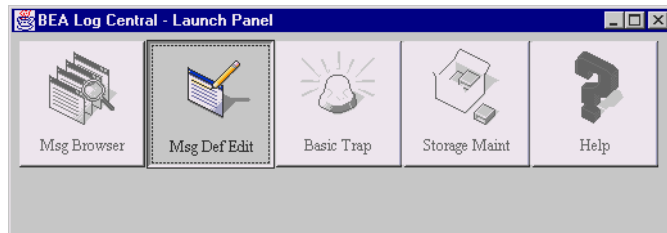
Document: Done

# Using the Message Definition Editor

Message definitions provide static information about messages that may be generated by various applications. These message definitions are described in Chapter 5, “Creating and Loading Message Definitions.” The Message Definition Editor allows you to create or modify message definitions one at a time. This is particularly useful if you already have an existing stock of definitions in the database and want to add a new definition on the fly.

To load many message definitions in a single action, you may find it more efficient to use the command-line utilities provided with Log Central. Messages are imported from an ASCII text file using the command-line utility `msgdef_import`. Message definitions in the text file must conform to the format used in the message definition template (`msgdef.template`) provided with Log Central. Messages can also be exported from the database to an ASCII text file using the command-line utility `msgdef_export`. For more information on these utilities, refer to Chapter 5, “Creating and Loading Message Definitions.”

Invoke the Message Definition main window by selecting the Msg Def Edit button from the Launch Panel.



The subjects discussed in this section are as follows:

- ◆ Message Definition Main Window
- ◆ How to Display Message Definitions
- ◆ How to Add a New Message Definition
- ◆ How to Modify a Message Definition
- ◆ How to Delete Message Definitions

- ◆ How to Define Filtering Criteria for Retrieving Message Definitions
- ◆ How to Change the Message Definition Layout
- ◆ How to Change Message Definition Colors
- ◆ How to Modify the Subsystem Description
- ◆ How to Generate Reports

## Message Definition Main Window

Figure 9-15 shows the basic look and presentation of the Message Definition main window.

**Figure 9-15 Message Definition Main Window**

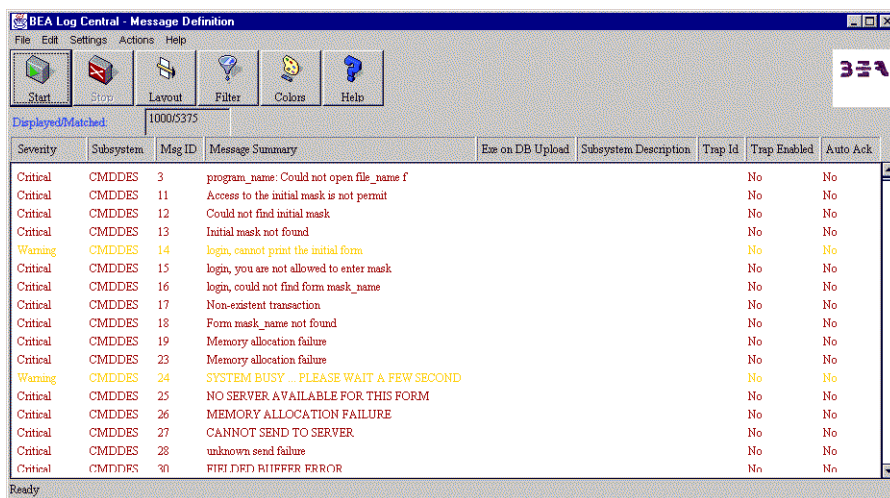




Figure 9-3 describes the fields of this window.

**Table 9-3 Message Definition Main Window Fields**

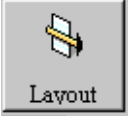



Field	Description
Menus	The File, Edit, Settings, Actions, and Help menus provide access to Message Definition Editor tools.
Toolbar	The toolbar buttons provides access to frequently used actions. Refer to Table 9-4 for a description of the toolbar buttons.
Displayed/Matched	When you execute a search of the database for message definitions matching a search filter, the main window displays the number of message definitions that match your search criteria and the number of message definitions displayed. These numbers differ if the search found more message definitions than the maximum that can be displayed. The maximum number of messages that can be displayed by the Message Browser is configurable, as detailed under “Invoking the Launch Panel.”
Message pane	Displays messages that meet the filtering criteria.

Table 9-4 describes the Message Definition toolbar buttons available for frequently used actions.

**Table 9-4 Message Definition Toolbar Buttons**

Tool Button or Field	Action
	Applies the settings defined in the Filter Settings window to select message definitions.
	Stops retrieval of message definitions. This button is grayed out if you have not yet started message definition retrieval.

**Table 9-4 Message Definition Toolbar Buttons**

Tool Button or Field	Action
 <p>Layout</p>	<p>Invokes the Layout Definition window. You can use this window to specify which fields in message definitions to display and which order to place the columns in the table.</p>
 <p>Filter</p>	<p>Invokes the Filter Settings window. You can use this window to:</p> <ul style="list-style-type: none"> <li>◆ Define criteria that are used to select the message definitions that are displayed in the main window.</li> <li>◆ Select the sort order for message definition display.</li> </ul>
 <p>Colors</p>	<p>Invokes the Message Colors window. You use this window to change the mapping of colors to message severity.</p>
 <p>Help</p>	<p>Invokes the help page.</p>

## How to Display Message Definitions

To display message definitions, select the Start button from the Message Definition main window. This brings up a list of message definitions in the scrollable display window, according to the filtering criteria. For more information, see “How to Define Filtering Criteria for Retrieving Message Definitions.”

## How to Add a New Message Definition

1. Select File → New Message Definition.

This brings up the Message Definition Details window, shown in Figure 9-16.

Alternatively, you can double-click on any message definition, and then select the File → New menu option.

**Figure 9-16** Message Definition Details Window

The screenshot shows a Java applet window titled "Message Definition Details". It features a "File" menu at the top left. The main area contains several form fields: "Subsystem Name" (a pull-down menu with "LIBTMIB -" selected), "Message Summary", "Execute on Upload", "Severity" (a pull-down menu with "Informational" selected), "Auto Ack" (a pull-down menu with "Yes" selected), "Message ID", "Trap ID", and "Trap Enabled" (a pull-down menu with "Yes" selected). Below these are two large text areas for "Description" and "Recommendation". At the bottom are three buttons: "Add", "Cancel", and "Help". A warning bar at the very bottom reads "Warning: Applet Window".

2. Select the subsystem name from the pull-down list, and continue filling out the fields as you want.

The Subsystem Name pulldown displays a list of subsystems in the Log Central database. To add a new subsystem entry, use the `subsystem_create` command, described in Chapter 3, “Configuring the Database for Use with Log Central.”



If you want to have a script or file execute when the Central Collector loads the specified message into the Log Central database, specify it in the Execute on Upload field.

Figure 9-17 shows a filled-out form.

**Figure 9-17 Message Definition Details Form Filled Out**

**Message Definition Details**

File

Subsystem Name: TUXEDO - BEA Tuxedo System

Message Summary: Memory allocation failed for compression

Execute on DB Upload: winssendalert.exe

Severity: Critical

Auto Ack: Yes

Message ID: 1206

Trap ID: 47

Trap Enabled: Yes

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 is allocated to the program.

Add Cancel Help

3. Select Save, and your definition is saved to the Log Central database.

## How to Modify a Message Definition

From the Message Definition main window, you can modify an existing message definition:

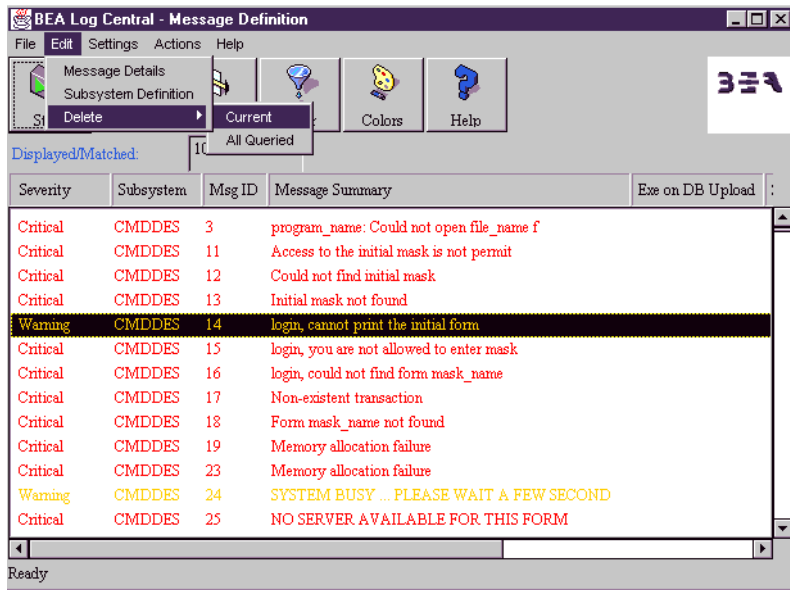
1. Double-click on the message definition that you want to modify.  
The Message Definition Details window appears with the fields filled in for the specific message definition.
2. Change these fields as you want.
3. Select Save, and your modified definition enters the message definition list.

## How to Delete Message Definitions

From the Message Definition main window, you can delete a message definition in two ways, as follows:

1. Select the message definition that you want to delete.
2. Select Edit → Delete from the Message Definition main window.
3. Select either Current or All Queried from the submenu, as shown in Figure 9-18.

**Figure 9-18 Message Definition Window: Delete**



Alternatively:

1. Double-click on the message definition that you want to delete.  
The Message Definition Details window appears.
2. Select File → Delete from the Message Definition Details window.

## How to Define Filtering Criteria for Retrieving Message Definitions

You can use filters to specify selection criteria to define which message definitions are selected for display.

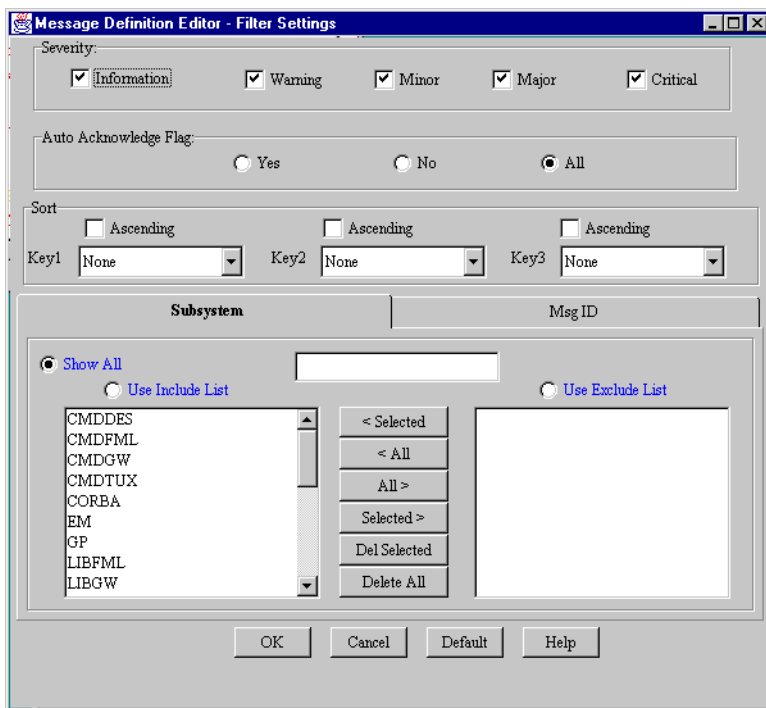
From the Message Definition main window, you can filter message definitions as follows:

1. Select the Filters button on the Message Definition main window.

Alternatively, you can select Settings → Editor Filter menu option.

The Message Definition Filter Settings window appears on the screen, as shown in Figure 9-19.

**Figure 9-19 Message Definition Filter Settings Window**



2. Define the selection criteria from which to view message definitions in the Message Definition Editor.
3. Select the sort order for message definitions. You can nest up to three sorts, mixing ascending and descending sort order.
4. If you want to specify different filtering criteria for either Subsystem or Message ID, select that tab.

Operations in these tabs are similar to those discussed under “How to Filter Messages,” in the Message Browser section.

5. Select either Use Include List or Use Exclude List.

For Use Include List, the specified features are used to determine which message definitions to display. For Use Exclude List, those message definitions that meet the selected criteria are specifically excluded from the display.

6. When you have made as many specifications as you want, select OK.

## How to Change the Message Definition Layout

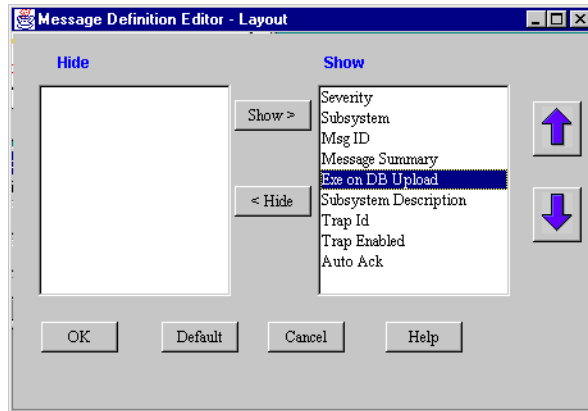
From the Message Browser main window, you can change the layout to define which message definition attributes are displayed in the main window or to change the order of the attributes in the tabular display:

1. Select Layout from the main window.

Alternatively, you can select the Settings → Layout Definition menu option.

The Message Definition Editor Layout window appears on the screen, as shown in Figure 9-20.

**Figure 9-20 Message Definition Editor Layout Window**



2. Select the components that you want to show or hide

You can move components into the appropriate column by clicking the Show or Hide arrows on the main Layout window. You can move components up and down with the blue arrows to the right. The order you specify in the Show panel determines the order these columns appear in the message panel of the Message Browser main window, shown in Figure 9-15.

3. Select OK

**Note:** If you want these selections as your default settings, select Default on the Layout window.

## How to Change Message Definition Colors

Message definitions appear in the Message Definition main window in different colors, dependent on severity. For example, you might want message definition of critical severity to appear in red, warnings in yellow, and so on. The default color-mapping is as follows.

Severity	Color
Informational	Green
Warning	Yellow
Minor	Blue
Major	Magenta
Critical	Red

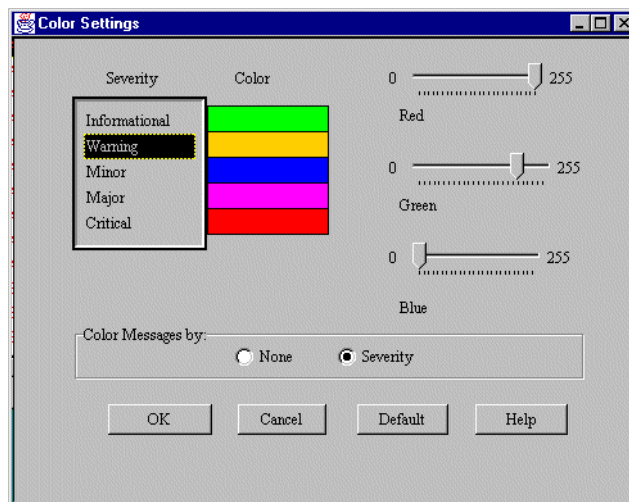
To change the mapping of colors to indicate message severity level:

1. Select Colors from the Message Definition main window.

Alternatively, you can select the Settings → Message Colors menu option.

The Color Settings window appears on the screen, as shown in Figure 9-21.

**Figure 9-21 Message Definition Editor Color Settings Window**



2. In the Severity column, select the severity level whose color you want to change.
3. Move the RGB sliders to get the color you want.

As you move the sliders, the color in the Color column changes.

4. Repeat steps 2 and 3 for as many severity levels as you want to change.

**Note:** If you do not want to map colors to severity levels, select the None radio button.

5. Select OK.

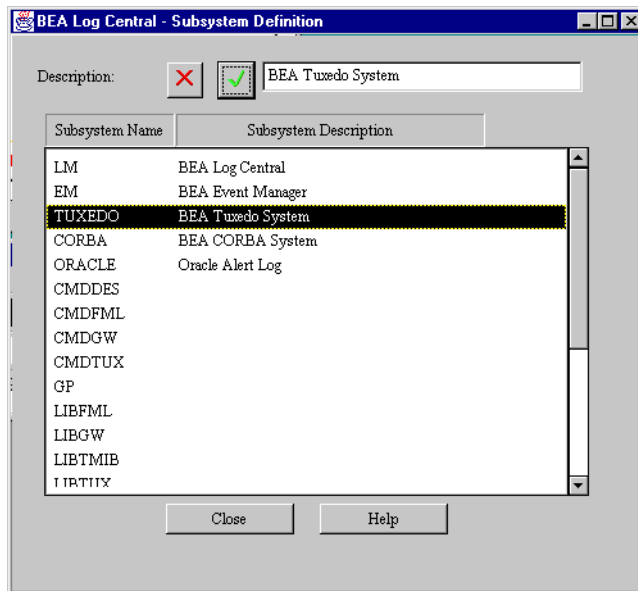
## How to Modify the Subsystem Description

You can modify the subsystem description associated with a subsystem name:

1. Select Edit → Subsystem Definition from the Message Definition main window.

The Subsystem Definition window appears on the screen, as shown in Figure 9-22.

**Figure 9-22 Subsystem Definition Window**



2. In the lower pane of the Subsystem Definition window, select the subsystem whose description you want to change.
3. Make the change you want in the description field.



4. Select the green checkmark to bring about the change.  
The red X undoes the change.
5. When you have made as many changes as you want, select Close.

## How to Generate Reports

You can generate two types of reports from the Message Definition Editor:

- ◆ Summary reports
- ◆ Detail reports

For both, the Message Definition Editor uses the current filter criteria to select messages on which to generate a report. The generated report is an HTML file created in *install\_dir*\web\applet (*install\_dir*/web/applet on UNIX), where *install\_dir* is the directory in which you installed Log Central. Viewing this file requires a Web browser. Make sure that you have made a Web browser accessible to the Console. For information on how to do this, refer to “Invoking the Launch Panel.”

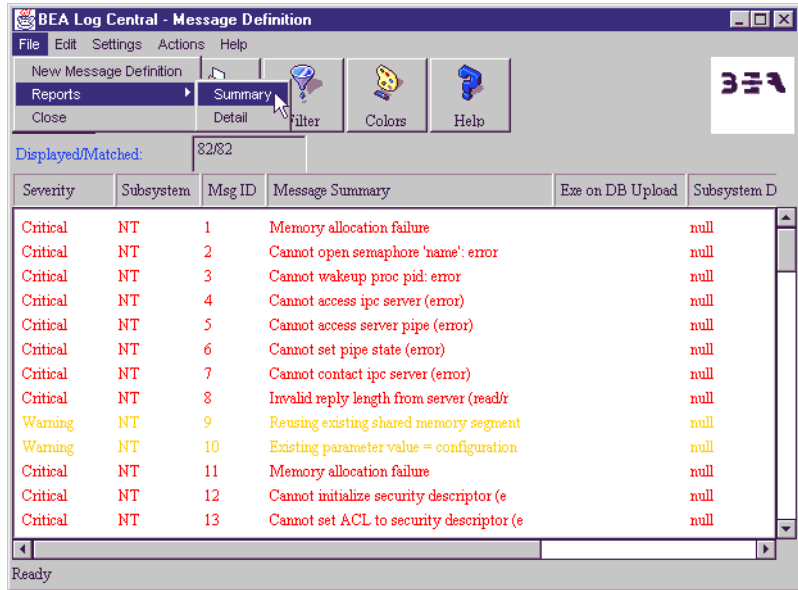
## How to View Summary Reports

To view summary reports in the Message Definition Editor, perform the following steps:

1. Specify the filtering criteria, as described in “How to Define Filtering Criteria for Retrieving Message Definitions.”
2. Select the Start button in the Message Definition main window.

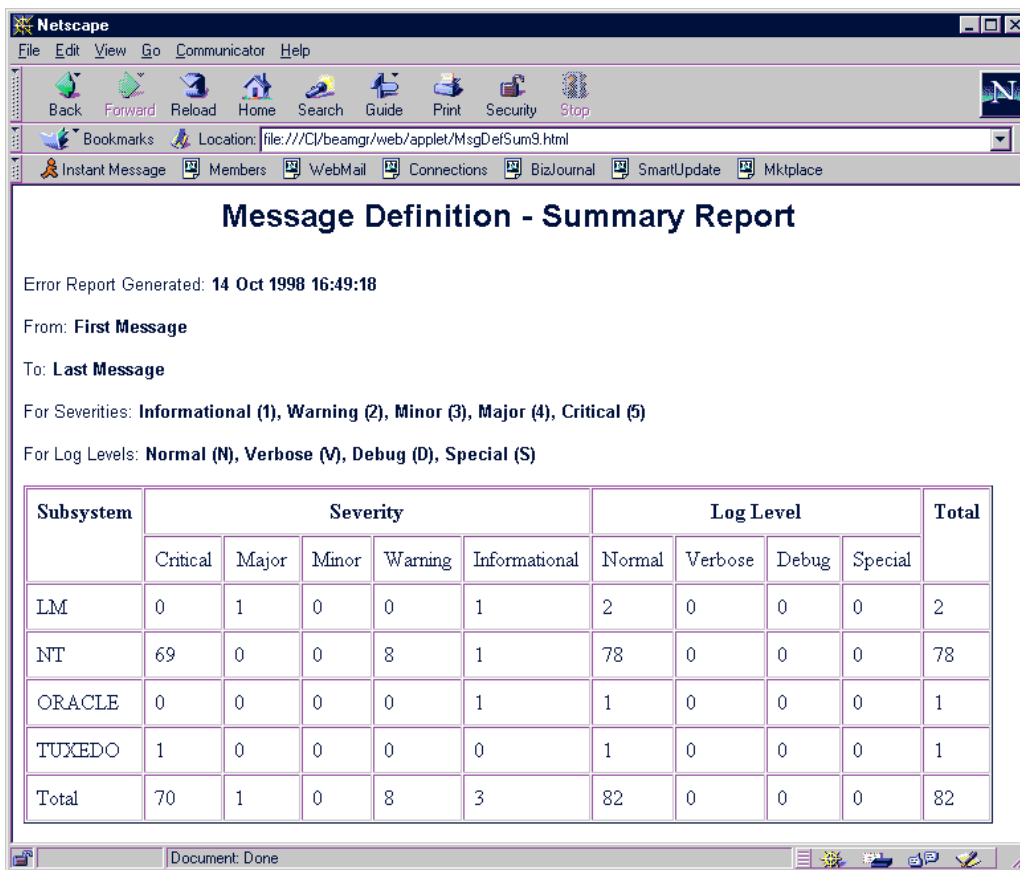
3. Select File → Reports → Summary, as shown in Figure 9-23.

**Figure 9-23 Message Definition Window, Choosing Summary Report**



A Web browser window appears, displaying the requested report, as shown in Figure 9-24.

**Figure 9-24 Message Definition Summary Report**

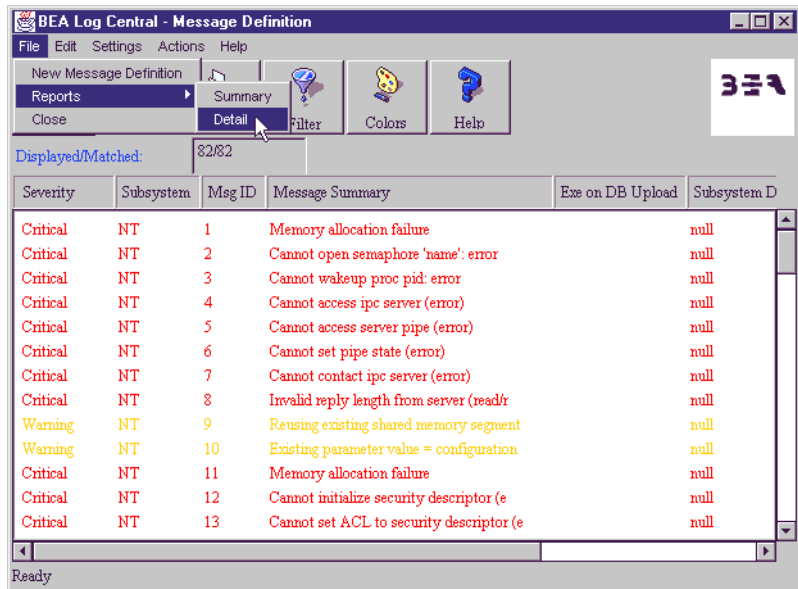


## How to View Detail Reports

To view detail reports in the Message Definition Editor, perform the following steps:

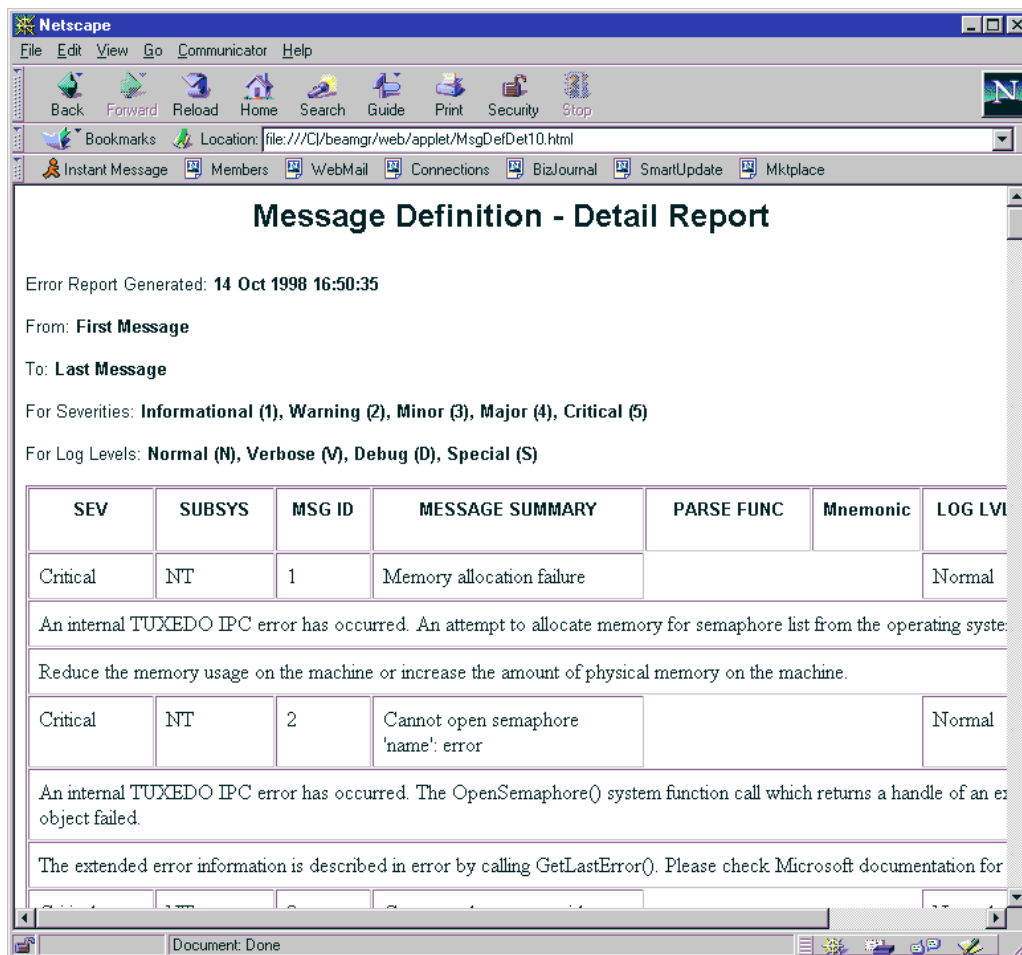
1. Specify the filtering criteria, as described in “How to Define Filtering Criteria for Retrieving Message Definitions.”
2. Select the Start button in the Message Definition main window.
3. Select File → Reports → Details, as shown in Figure 9-25.

**Figure 9-25 Message Definition Window, Choosing Details Report**



A Web browser window appears, displaying the requested report, as shown in Figure 9-26.

**Figure 9-26 Message Definition Details Report**



# Using the Basic Trap Configuration Tool

You can configure the Central Collector to send SNMP trap notifications based on one (or both) of the following attributes in the message definition for the incoming message:

- ◆ Severity
- ◆ Trap Enabled

If you enable trap generation based on the Trap Enabled field in the message definition, a trap is generated for any message whose definition has Trap Enabled set to YES.

Invoke the Basic Trap Configuration window by selecting the **Basic Trap** button from the Launch Panel.



The subjects discussed in this section are as follows:

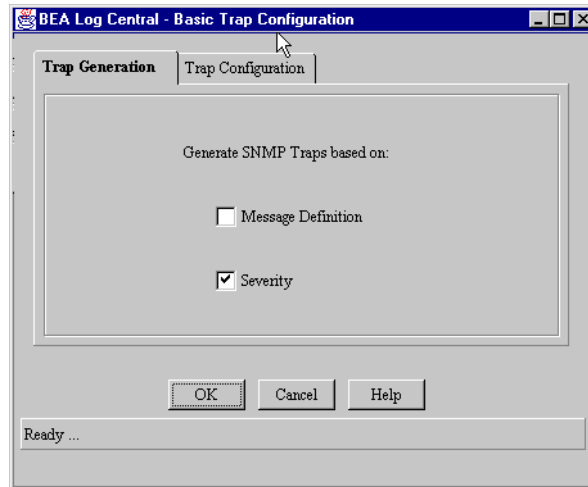
- ◆ Configuring Trap Generation by Severity
- ◆ Configuring Trap Generation by Message Definition
- ◆ Generating Traps Based on both Severity and Message Definition

## Configuring Trap Generation by Severity

To enable trap generation by severity, do the following:

1. Invoke the Basic Trap Configuration window from the Log Central Console Launch Panel.
2. Select the Trap Generation tab (if it is not already selected) on the Basic Trap Configuration window and select Severity. The Trap Generation tab, shown in Figure 9-27, has checkboxes for severity and message definition to determine which to use as the basis of trap generation.

**Figure 9-27 Basic Trap Configuration, Trap Generation Tab**



3. Select the Trap Configuration tab.
4. Select which severities will generate traps.  
If you choose to enable any severity listed on this page, then any message having that severity prompts an SNMP trap.
5. Use the ID of Resulting Trap Field to assign an enterprise-specific trap number to traps for the selected severities.

6. When you have filled out the fields of this tab, select OK.

Figure 9-28 shows an example of the Trap Configuration tab filled out.

**Figure 9-28 Basic Trap Configuration, Trap Configuration Tab**

The screenshot shows a dialog box titled "BEA Log Central - Basic Trap Configuration". It has two tabs: "Trap Generation" and "Trap Configuration", with the latter being the active tab. Inside the "Trap Configuration" tab, there is a table with three columns: "Message Severity Encountered", "Trap Enabled", and "ID of Resulting Trap". The table contains five rows for different severity levels: Informational, Warning, Minor, Major, and Critical. Each row has a checked checkbox in the "Trap Enabled" column and a text box in the "ID of Resulting Trap" column containing a number. At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help". A status bar at the very bottom says "Ready ...".

Message Severity Encountered	Trap Enabled	ID of Resulting Trap
Informational	<input checked="" type="checkbox"/>	2081
Warning	<input checked="" type="checkbox"/>	2082
Minor	<input checked="" type="checkbox"/>	2803
Major	<input checked="" type="checkbox"/>	2804
Critical	<input checked="" type="checkbox"/>	2805

This enterprise-specific trap number is included in the SNMP trap packet sent for messages with the indicated severity. SNMP has no concept of the severity of an event and no special field is provided for severity in the SNMP trap packet. However, by assigning a different enterprise-specific number to the different severities, system administrators can map the Log Central traps to severities on their enterprise-management system.



## Configuring Trap Generation by Message Definition

To enable trap generation by message definition, do the following:

1. Invoke the Basic Trap Configuration window from the Log Central Console Launch Panel.
2. Select the Trap Generation tab (if it is not already selected) and select Message Definition. The Trap Generation tab has checkboxes for severity and message definition to determine which to use as the basis of trap generation.
3. Select OK.

If you enable trap generation based on Message Definition, then a trap is generated for an incoming message if the Trap Enabled field in the message definition for that message is set to **YES**. If you want to ensure that the Trap Enabled field for a given message type is set to **YES**, invoke the Message Definition Editor and view the message definition and modify the Trap Enabled field if necessary. The message definition also defines the specific trap number that is sent in the trap packet when a trap is generated based on the Trap Enabled field being set to **YES**.

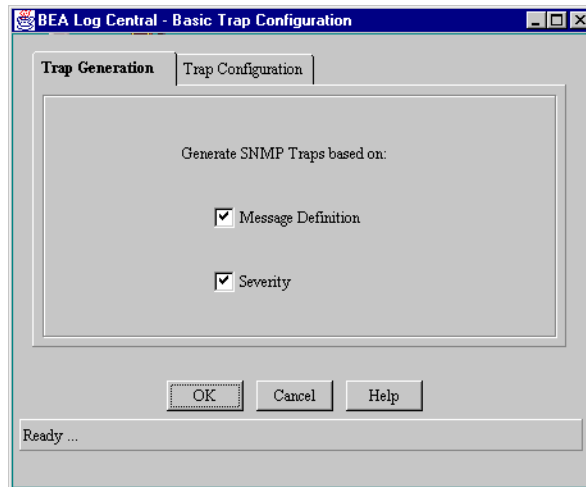
## Generating Traps Based on both Severity and Message Definition

You can configure the Central Collector to send SNMP traps based on both severity and message definition. In that case, you do not need to set the ID of Resulting Trap field on the Trap Configuration tab (Step 5 under “Configuring Trap Generation by Severity”) because the specific trap number is determined by the message definition.

To enable trap generation by message definition, do the following:

1. Invoke the Basic Trap Configuration window from the Log Central Console Launch Panel.
2. Select the Trap Generation tab (if it is not already selected) and select both Message Definition and Severity, as shown in Figure 9-29.

**Figure 9-29** Trap Generation Tab: Severity and Message



3. Select OK.

If you select both severity and message definition as the basis of generating traps, traps are generated for messages with the selected severities, but only if the message definition Trap Enabled field is set to **YES**.

# Using the Storage Maintenance Tool

The Log Central Central Collector creates several intermediate files (one intermediate file per hour) that need easy management and pruning. The number of messages in the database can also grow rapidly. The Storage Maintenance tool eases the management of these files and messages, allowing you to set up periodic processing and deletion of files and messages.

Invoke the Storage Maintenance main window by selecting the Storage Maint button from the Launch Panel.



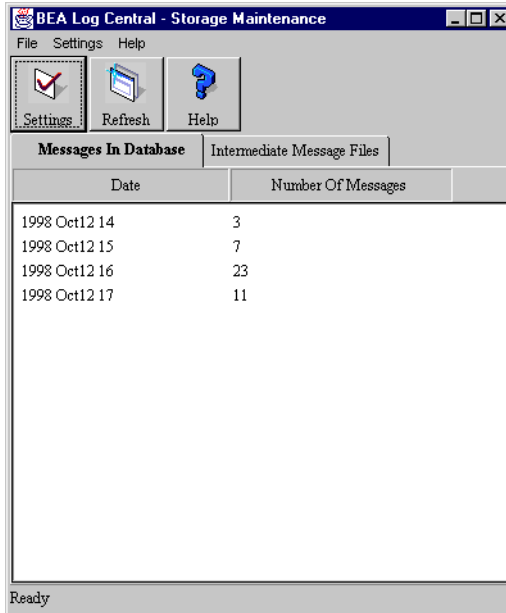
The subjects discussed in this section are as follows:

- ◆ Storage Maintenance Main Window
- ◆ How to Prepare the Records Processing Script
- ◆ How to Schedule Deletion of Database Records
- ◆ How to Manually Process Database Records
- ◆ How to Schedule Processing of Database Records
- ◆ How to Schedule Both Processing and Deletion of Database Records
- ◆ How to Schedule Deletion of Intermediate Files

## Storage Maintenance Main Window

When the Storage Maintenance main window first appears, the Messages in Database tab is selected, as shown in Figure 9-30.

**Figure 9-30** Storage Maintenance Main Window



This displays a representation of the number of messages delivered during each hour of the day. The date appears on each line in the following format:

YYYY MMDD HH

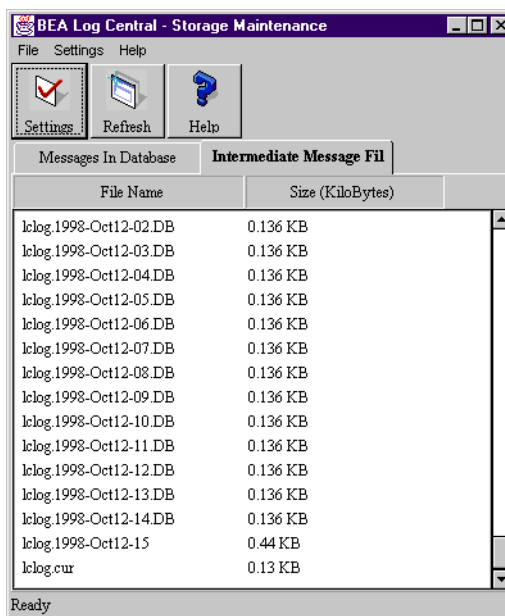
As, for example:

1998 Oct12 14

which means during the 2 to 3 p.m. time period on October 12. (The hour is in 24-hour representation.) During that hour, three messages were stored in the database.

Figure 9-31 shows the Storage Maintenance main window with the Intermediate Message Files tab selected.




**Figure 9-31 Storage Maintenance Main Window, Intermediate Message Files Tab**



Each file name represents one hour of the day. The file name consists of the intermediate file prefix, plus a date and time representation similar to the Messages in Database format. For example, the file name corresponding to 1998 Oct12 14 might be `lclog.1998-Oct12-14.DB`. The current file is called `lclog.cur`.

The toolbar is below the menu bar. Table 9-5 describes the Storage Maintenance toolbar buttons available for frequently used actions.

**Table 9-5 Storage Maintenance Operations Settings Toolbar Buttons**

Tool Button or Field	Action
	Invokes the Operations Settings configuration window.
	Refreshes the display, making it consistent with the records in the database and intermediate files.
	Invokes the help page.

## How to Prepare the Records Processing Script

You can have Log Central execute a script of your choosing on its intermediate files and on messages in the database. Log Central includes a sample script that you modify for your purposes.

To tailor this script to your specifications, perform the following steps:

1. With an ASCII text editor, open the file `install_dir\bin\process_dbrec.bat` (`install_dir/bin/process_dbrec` on UNIX), where `install_dir` is the directory in which you installed Log Central.
2. Modify the script according to the description following these steps.
3. Save the script.

The `process_dbrec` script ships as a sample file—a shell script on UNIX machines or a batch file on Windows NT—with comments describing how to use it (but it does not actually do anything as shipped). The Operations Settings window allows you to specify how frequently to execute the script, or whether to even execute it at all. You can also process selected records manually using the Process Records option of the File

menu. The script is invoked for the selected database records when you select this option. As a Log Central administrator, you will want to tailor this file to your specifications.

Some of the things you could do using the `process_dbrec` script include:

- ◆ Generate an email message that indicates which messages were deleted and the date and time of the operation
- ◆ Copy the archive file to a tape
- ◆ Use an SQL statement to move specified records into a different table in the database

The `process_dbrec` command uses the following syntax:

```
process_dbrec starttime endtime
```

The `process_dbrec` options definitions follow.

Argument	Description
<i>starttime</i>	Start time. Specifying 0 here means process all records before <i>endtime</i> .
<i>endtime</i>	End time.

Your program can use these parameters as needed. It could also ignore them entirely.

Log Central expects your program to return -1 if there is an error and 0 if it runs successfully. Only if the return value is 0 are records deleted after the `process_dbrec` script runs, even if you have selected the Delete Records option.

**Note:** Do not put any commands into the `process_dbrec` script that write to standard out (such as `echo`). The standard out terminal is not available in this context.

## Example

Based on the specifications you made in the Operations Settings window, Log Central might execute the script as follows:

```
process_dbrec "3-FEB-98 11:00" "4-MAR-98 19:00:00"
```

## How to Schedule Deletion of Database Records

You can have the Storage Maintenance tool delete records from the Log Central database according to a schedule you determine by performing the following steps:

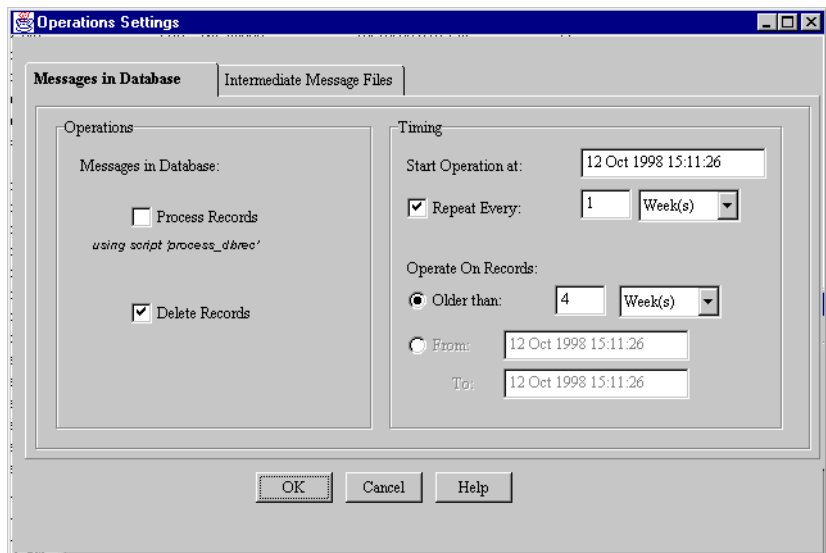
1. Select Settings on the toolbar.

This brings up the Operations Settings window, with the Messages in Database tab selected.

Alternatively, you can select the Settings → Operations menu option.

2. Select only the Delete Records checkbox, as shown in Figure 9-32.

**Figure 9-32 Storage Maintenance Main Window, Delete Records**



3. Specify the start time of the operation.
4. Specify the repetition time of the operation.

If you want to perform the operation once only, deselect the Repeat Every checkbox.



5. Specify the interval of the operation.

Enter a number in the first field, and then select a time unit from the pulldown menu in the second field.

6. Select the records to be deleted, in one of two ways:

- ◆ You can select a moving window of records by specifying Older Than in the Operate on Records section.

As in step 5, enter a number in the first field, and then select a time unit from the pulldown menu in the second field. If you choose, for example, Older Than 2 weeks, and your operation is scheduled weekly, a record that is 2 weeks and 1 hour old at the time the operation is invoked would not be deleted, but it would be the next time.

- ◆ You can select a fixed window of records by specifying a From and To time in the Operate on Records section.

## How to Manually Process Database Records

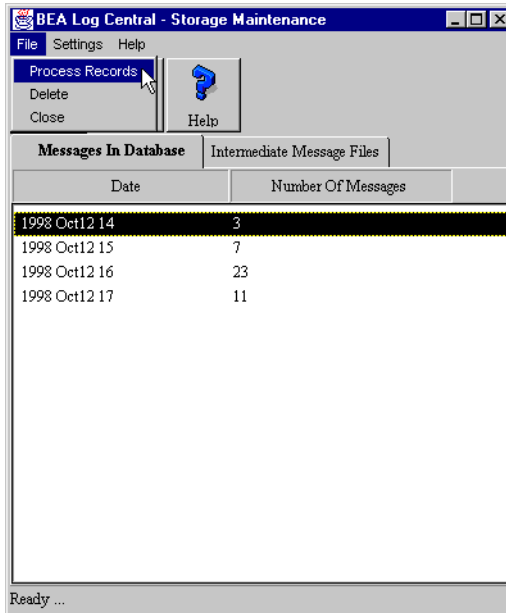
You can have the Storage Maintenance tool execute the `process_dbrec` script on individually selected records on a “one-shot” basis by performing the following steps:

1. Select the records to process within the scrollable display in the Messages in Database tab.

This operation is not valid in the Intermediate Message Files tab.

2. Select File → Process Records, as shown in Figure 9-33.

**Figure 9-33 Storage Maintenance Main Window, Process Records**



This causes the `process_dbrec` script to be executed immediately on the selected records.

## How to Schedule Processing of Database Records

You can have the Storage Maintenance tool execute the `process_dbrec` script on selected records according to a schedule you determine by performing the following steps:

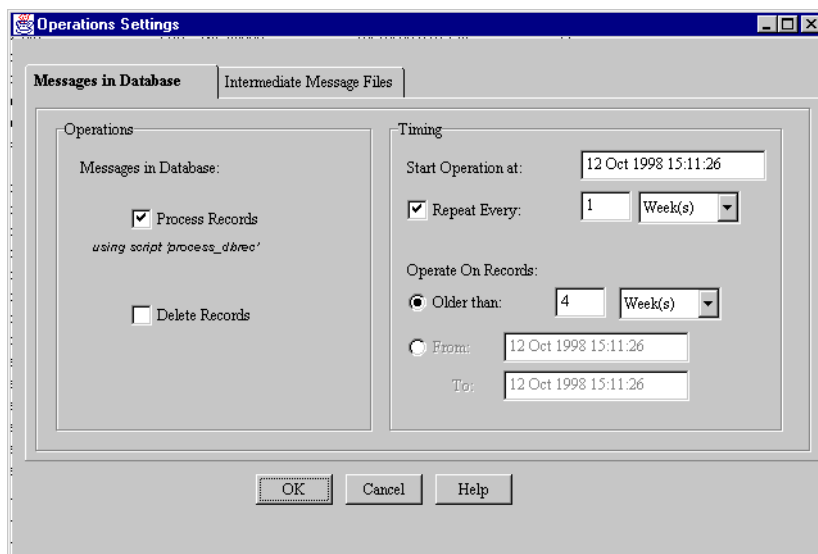
1. Select Settings on the toolbar.

This brings up the Operations Settings window, with the Messages in Database tab selected.

Alternatively, you can select the Settings → Operations menu option.

2. Select only the Process Records checkbox, as shown in Figure 9-34.

**Figure 9-34 Operations Settings Window, Records Processing**



3. Specify the start time of the operation.

4. Specify the repetition time of the operation.

If you want to perform the operation once only, deselect the Repeat Every checkbox.

5. Specify the interval of the operation.

Enter a number in the first field, and then select a time unit from the pulldown menu in the second field.

6. Select the records to be processed, in one of two ways:

- ◆ You can select a moving window of records by specifying Older Than in the Operate on Records section.

As in step 5, enter a number in the first field, and then select a time unit from the pulldown menu in the second field. If you choose, for example, Older Than 2 weeks, and your operation is scheduled weekly, a record that is 2 weeks and 1 hour old at the time the operation is invoked would not be deleted, but it would be the next time.

- ◆ You can select a fixed window of records by specifying a From and To time in the Operate on Records section.

## How to Schedule Both Processing and Deletion of Database Records

You can have the Storage Maintenance tool execute the `process_dbrec` script on selected records, followed by deletion of those records from the Log Central database, according to a schedule you determine by performing the following steps:

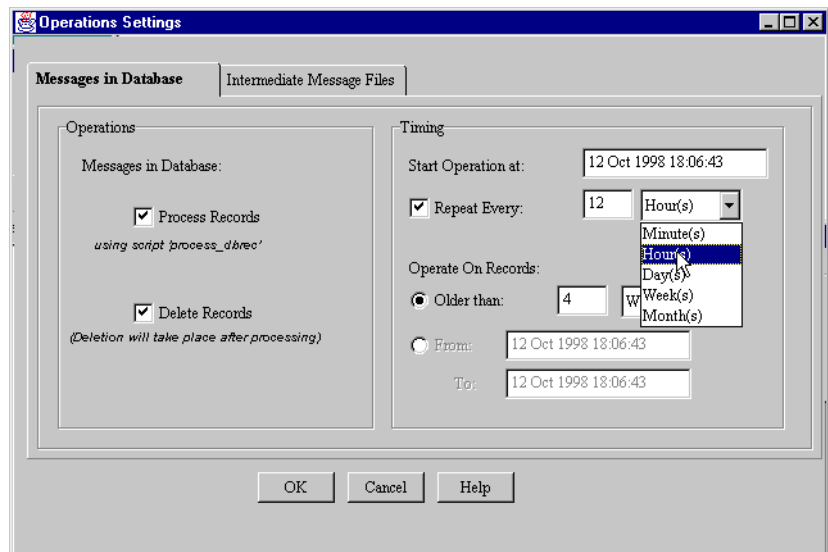
1. Select Settings on the toolbar.

This brings up the Operations Settings window, with the Messages in Database tab selected.

Alternatively, you can select the Settings → Operations menu option.

2. Select both the Process Records and Delete Records checkboxes, as shown in Figure 9-35.

**Figure 9-35** Operations Settings Window, Records Processing and Deleting



3. Specify the start time of the operation.

4. Specify the repetition time of the operation.

If you want to perform the operation once only, deselect the Repeat Every checkbox.

5. Specify the interval of the operation.

Enter a number in the first field, and then select a time unit from the pulldown menu in the second field.

6. Select the records to be processed, in one of two ways:

- ◆ You can select a moving window of records by specifying Older Than in the Operate on Records section.

As in step 5, enter a number in the first field, and then select a time unit from the pulldown menu in the second field. If you choose, for example, Older Than 2 weeks, and your operation is scheduled weekly, a record that is 2 weeks and 1 hour old at the time the operation is invoked would not be deleted, but it would be the next time.

- ◆ You can select a fixed window of records by specifying a From and To time in the Operate on Records section.

## How to Schedule Deletion of Intermediate Files

You can have the Storage Maintenance tool delete intermediate files according to a schedule you determine by performing the following steps:

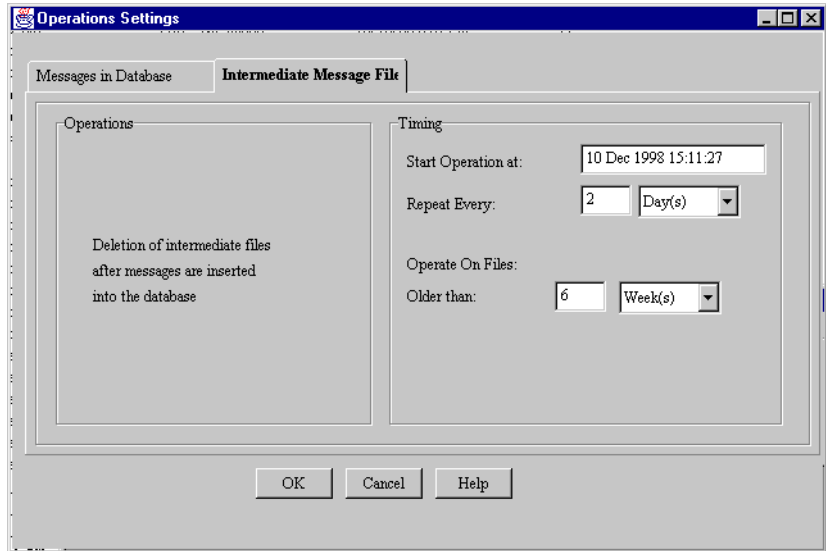
1. Select Settings on the toolbar.

This brings up the Operations Settings window, with the Messages in Database tab selected.

Alternatively, you can select the Settings → Operations menu option.

2. Select the Intermediate Message Files tab, as shown in Figure 9-36.

**Figure 9-36 Operations Settings Window, Intermediate Message Files Tab**



3. Specify the start time of the operation.
4. Specify the interval of the operation.  
Enter a number in the first field, and then select a time unit from the pulldown menu in the second field.
5. Select the age of files to be deleted by specifying Older Than.  
As in step 4, enter a number in the first field, and then select a time unit from the pulldown menu in the second field.

# A Message Format

All Log Central messages use a common format. This appendix discusses the following topics:

- ◆ Message Format
- ◆ Message Attributes at the Agent
- ◆ Attributes in the Message Definition

# Message Format

A message consists of the following components:

- ◆ Header, containing predetermined fields of data.
- ◆ Body, containing information defined by application developers.
- ◆ Attributes in the message type definition. The message type definition is stored in the Log Central relational database.

The Header and Body are the message components that are available at the data collection agent. When data collection agents extract information from messages flowing into logs on the managed node, the content of each message header is determined by the mapping of log messages into Log Central message format. Each message is uniquely identified by two of the attributes in the header: Subsystem Name and a Message ID within that subsystem.

In addition to the attributes in the header, additional attributes are attached to a log message at the Central Collector. These additional attributes are those contained in the message definition, which is stored in the Log Central database. These additional attributes include the following:

- ◆ Severity
- ◆ Description
- ◆ Recommendation
- ◆ Trap Generation
- ◆ Trap ID
- ◆ Summary
- ◆ Automatic Acknowledgment Flag
- ◆ Execute on DB Upload



# Message Attributes at the Agent

The following attributes are included in the log message as sent by a data collection agent.

## Log ID

The log ID uses a string of up to three characters to distinguish one set of log messages from another. For example, the IDs `IRX` and `DCS` might distinguish messages logged by a drug claim system from those logged by a document control system. This enables logically separate log information to be maintained in a single database.

The use of this field in Log Central messages is optional.

## Logging Level

This attribute consists of a single character. The possible values and their recommended interpretation are as follows:

- ◆ `N`—Normal message
- ◆ `D`—Debug message
- ◆ `V`—Verbose message
- ◆ `S`—Special message

## Date and Time

This string indicates the date and time on the host where the message originated (not the date and time on the host where the log file resides). The format (showing month, day, hour, minute, second, and year) is:

*Mmm dd hh:mm:ss yyyy*

## Subsystem Name

This string is the component of the system that logs the message. The subsystem name field presupposes that your software is functionally divided into subsystems. The subsystem name must be unique throughout the entire network to identify a functional group.

## Message ID

The Message ID is a number in the range of 1 to 99999. This attribute identifies the type of message within a subsystem (i.e., it is unique within a subsystem, and two subsystems may include the same message ID). The message ID and the Subsystem Name together uniquely identify a message. The convention for message ID assignments is as follows.

Message ID Number	Purpose
1 - 49999	Activity and error messages defined by application developers.
50000 - 89999	Reserved for future use.
90000 - 99999	General purpose system management messages, defined in the Log Central header file. Specifically: 90000 – Startup message 90001 – Shutdown message 90002 – Transaction processing log message 90100 – Transaction processing log message (NCPDP) 90201 - 99999 – Reserved for future use

## Host Name

This is the name of the network host where the message originated.

## Process ID

This is the numeric process identifier (PID) of the process that issued the message.

## User ID

This is the user ID of the process that issued the message.

## Function Name

This attribute is optional. This is the name—up to 40 characters in length—of the internal function that issued the message. If this attribute is included, the recommended convention is:

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

## Transaction ID

This optional 21-character string attribute helps to correlate a message with other error messages logged during the same transaction, or with data saved in another relational database supporting an OLTP system. This field is relevant only in a case in which the application process was operating in the context of a transaction.

Log Central represents a nonexistent transaction ID with a 0 value.

## Body

The message body contains free-format information in a text format. The content of this string is determined by the developer of the application that logs the message. The maximum length of a message body is 2000 characters.

# Attributes in the Message Definition

The following are attributes of the message definition. The message definition can be modified using the Message Definition Editor. For more information, refer to Chapter 9, “Using the Log Central Console.”

## Severity

Definitions of message types stored in the Log Central database include a message classification. The categories used to classify messages are defined by the application developer but a typical use is to categorize messages by severity.

The severity of a message is a rating used to represent the importance or impact of an event. For example, a message that indicates high usage of a print spooler reports a less severe event than a message telling you that an application server has crashed. The Log Central Central Collector assigns a severity to messages as it saves the message in the Log Central database or to generate SNMP traps. The Central Collector uses the severity classification that has been included in the message definition. By default, Log Central messages use the standard ISO severities, described in the following table.

Severity	Description
Informational	This includes all activity messages.
Warning	These messages report a problem that does not need immediate attention.
Minor	Performance or service degradation may result.
Major	The problem needs immediate attention or the performance of the system may deteriorate.
Critical	The software detected an error that requires immediate attention. The system is down or may soon go down.

## Summary

This field gives a summary of the information in the Description field.

## Description

This attribute is a description of the condition or event that the message is reporting, such as the probable cause of a problem. The message description is contained in the message definition, not in the message header. This attribute can be modified using the Log Central Message Definition Editor.

## Recommendation

This attribute is the recommended action to be taken when this message occurs, that is, an action that probably solves the problem. This attribute is contained in the message definition, not in the message packet that is generated by the data collection agents. You can modify this attribute, based on your past experience and particular configuration, to provide enhanced advice for future situations when the same message recurs. Use the Message Definition Editor to modify the recommendation field. For more information on using the Log Central Console, refer to Chapter 9, “Using the Log Central Console.”

## Trap Generation

The value of this attribute is either YES or NO. The Basic Trap Configuration window of the Log Central Console allows you to instruct the Central Collector to generate SNMP trap notifications. If you specify that the message definition is to be used to select which messages trigger a trap, traps are generated if the value of this attribute is set to YES. For more information on using the Log Central Console, refer to Chapter 9, “Using the Log Central Console.”

### **Trap ID**

You can use the Basic Trap Configuration window of the Log Central Console to configure the Central Collector to generate SNMP trap notifications. If you specify that the message definition is to be used to select which messages trigger a trap, the value of the Trap ID attribute is used as the Specific Trap value in the enterprise-specific SNMP trap packet that is generated. For more information on using the Log Central Console, refer to Chapter 9, “Using the Log Central Console.”

### **Automatic Acknowledgment Flag**

When messages arrive at the Central Collector, they are, by default, not acknowledged. You can change the value to `acknowledged` from the Message Browser. A typical use for this attribute is to chart which system problems are currently being resolved or actively investigated. If the automatic acknowledgment flag is set to `YES`, messages are automatically marked as acknowledged. Possible values are `NO` or `YES`.

### **Execute on DB Upload**

If a value is set for this field, the specified program or script is executed when the live message is saved into the database.

# B Environment Variables

Under certain conditions, you may need to change some of Log Central's environment variables. This appendix describes these modifiable Log Central environment variables, and the conditions under which you need to modify them:

- ◆ `BEA_LC_IPCKEY`
- ◆ `BEA_LC_CONF_SERVICE`
- ◆ `BEA_SM_BEAMGR_CONF`

## BEA\_LC\_IPCKEY

The `BEA_LC_IPCKEY` environment variable defines the interprocess communication (IPC) key used by Log Central, as well as acting as an identifier for Log Central.

If you are using only one instance of Log Central and have no need of changing its default IPC key, you do not need to modify this variable. You must set it for multiple Log Central systems because each needs a different value for its IPC key. The default value of `BEA_LC_IPCKEY` is `0xeeee0000`. You can use the default for the first instance of Log Central, but you must change the variable in any succeeding invocations of Log Central. The value you set must be the same as that in the `IPCKEY` keyword of the `LC_GLOBAL` entry in the `messaging.conf` configuration file.

`BEA_LC_IPCKEY` is used by the following Log Central processes:

- ◆ `stop_messaging`
- ◆ `show_config`
- ◆ `log_monitor`

- ◆ msg\_test
- ◆ start\_messaging

The start\_messaging process on the central host reads the value of its IPC key from the messaging.conf configuration file (from the IPCKEY keyword). The value is then passed to all processes started by start\_messaging (ipc\_config, proc\_monitor, msg\_sender, msg\_receiver, msg\_processor). All other Log Central programs not started by start\_messaging (such as log\_monitor, msg\_test) read the value of this variable from the environment. The start\_messaging process on a managed node also reads the value from the environment.

You should set this value on the central host as well as all managed nodes before starting any Log Central processes.

## Using a New IPCKEY Value

The following steps show how to assign a different value from the default to the IPCKEY.

1. Set the value of the IPCKEY keyword of the LC\_GLOBAL entry in the messaging.conf configuration file on the central host. (IPCKEY can take any numeric value. Use 0x to indicate a hexadecimal value.) For example, to use 0xe1e1e1e1, you might have an LC\_GLOBAL entry like the following:

```
LC_GLOBAL
{
    CENTRAL_HOST      = "quahog"
    LOGPREFIX         = "/usr/lclog"
    BACKUP_HOST       = "orca"
    BACKUP_LOGPREFIX  = "/usr/backuplog"
    IPCKEY            = 0xe1e1e1e1
}
```

2. Set the BEA\_LC\_IPCKEY environment variable to the new value before starting the stop\_messaging, show\_config, log\_monitor, or msg\_test commands. Use the following command on UNIX with C-shell:

```
setenv BEA_LC_IPCKEY 0xe1e1e1e1
```

Use the following command on Windows NT, at a DOS prompt:

```
SET BEA_LC_IPCKEY 0xe1e1e1e1
```



# BEA\_LC\_CONF\_SERVICE

The BEA\_LC\_CONF\_SERVICE environment variable defines the name of the User Datagram Protocol (UDP) service used for communication between the `start_messaging` process on a managed node to the `start_messaging` process on the central host. The default value is `lc_conf`. This value is defined during installation. That value, whether `lc_conf` or some other name, must be added to the services database (`/etc/services` on UNIX) by the system administrator. If the administrator adds some value other than `lc_conf`, then you make that specification in the BEA\_LC\_CONF\_SERVICE environment variable.

You should set this value on the central host as well as all managed nodes before starting the `start_messaging` process.

## Using a Different UDP Communication Service

The following steps show how to assign a different value from the default to the communication service. These steps change from the default of `lc_conf` to `lc_conf_tux`.

1. Add `lc_conf_tux` as a UDP service in the services database (`/etc/services` on UNIX). For example:

```
lc_conf_tux      7011/udp
```

You may need the help of your systems administrator to make this change.

2. Set the BEA\_LC\_CONF\_SERVICE environment variable to the new value. Use the following command on UNIX with C-shell:

```
setenv BEA_LC_CONF_SERVICE lc_conf_tux
```

Use the following command on Windows NT, at a DOS prompt:

```
SET BEA_LC_CONF_SERVICE lc_conf_tux
```

## BEA\_SM\_BEAMGR\_CONF

The BEA\_SM\_BEAMGR\_CONF environment variable specifies the location of the BEA Manager configuration file that contains the TRAP\_HOST entry. TRAP\_HOST entries define the location of SNMP management stations that are configured to receive SNMP traps from Log Central.

On UNIX, the default location is `/etc/beamgr.conf`; on NT, the default location is `C:\etc\beamgr.conf`. If you wish to use a file location other than the default, you should set the BEA\_SM\_BEAMGR\_CONF environment variable to point to the other location.

You should set this value on the central host as well as all managed nodes before starting the `start_messaging` process.

## Setting a New Location for the BEA Manager Configuration File

The following steps show how to assign a different value from the default for the location of the BEA Manager configuration file. The BEA\_SM\_BEAMGR\_CONF environment variable must contain the absolute path of the file, including the name of the file.

- ◆ Use the following command on UNIX with C-shell:

```
setenv BEA_SM_BEAMGR_CONF /usr/home/myconfig.conf
```

- ◆ Use the following command on NT, at a DOS prompt:

```
SET BEA_SM_BEAMGR_CONF C:\usr\homemyconfig.conf
```

# C Testing and Debugging Commands

Log Central provides a number of testing and debugging utilities. This appendix describes how to:

- ◆ Generating Test Messages
- ◆ Reading the Current Intermediate Log File

## Generating Test Messages

Log Central provides a program called `msg_test` to test the flow of messages. You can observe the results of your test messages with the Message Browser of the Log Central Console. The `msg_test` utility generates a user-defined number of messages and sends them according to the settings specified in its options. You can also use `msg_test` to print performance data.

To generate test messages, run the `msg_test` command at a command prompt on the central host or a managed node, with the following syntax:

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

The `msg_test` command options definitions follow.

Argument	Description
<code>-h</code>	Displays a help message.
<code>-i</code>	Invokes the interactive mode. The <code>-i</code> option overrides the <code>-t</code> option.
<code>-l <i>length</i></code>	Defines the message body length. The default length is 40.
<code>-m <i>log_level</i></code>	Specifies one or more logging levels to use in the message header. <i>log_level</i> is one or more of the following: N – Normal (the default) V – Verbose D – Debug S – Special
<code>-n <i>messages</i></code>	Defines the number of messages you want to generate. The default is 1.
<code>-t <i>interval</i></code>	Defines the time interval, in seconds, to sleep between messages. The default is 0.
<code>-s <i>subsystem</i></code>	Specifies the subsystem name to use in the message header. You can specify more than one subsystem. The default is LC. You can supply a list of subsystem names.

## Examples

This command simply generates three random messages:

```
msg_test -n 3
```

The following two commands are equivalent; each generates three messages. The subsystem name of the first message is LC, the second is TUXEDO, and that of the third is LC (because it is not specified, and thus uses the default).

```
msg_test -n 3 -s "LC TUXEDO"
```

```
msg_test -n 3 -s LC -s TUXEDO
```

# Reading the Current Intermediate Log File

The `msg_reader` utility continuously reads the current intermediate log file that the `msg_receiver` process constructs, and writes the contents of this file to the standard output stream. The data appears in `stdout` in message format (that is, in a string containing header data and body text).

The `msg_receiver` process creates a new intermediate log file every hour. The `msg_reader` process automatically switches to reading the current log file, which the `msg_server` process updates hourly. It tracks the current log file, and automatically switches to a new log file whenever the `msg_server` closes an old log file and opens a new one. The `msg_reader` process displays the contents of the intermediate log file on the standard output, allowing you to monitor the file without going through the Console.

To read the current log file, run the `msg_reader` command at a command prompt on the central host, with the following syntax:

```
msg_reader [ -e ] [ -n ] [ -h ] pathname
```

The `msg_reader` command options definitions follow.

Argument	Description
<code>-e</code>	Starts writing from the end of the log file.
<code>-n</code>	Formats the logged messages before writing them. Do not use this option if you invoke the <code>msg_reader</code> along with the <code>log_monitor</code> application.
<code>-h</code>	Displays a help message.
<i>pathname</i>	Specifies the path and file name to open for reading. The <code>msg_reader</code> utility takes <i>pathname</i> as an argument, adds the extension <code>.cur</code> , and opens the resulting file for reading. The <i>pathname</i> value is the intermediate file prefix for Log Central.

### Examples

The following command invokes `msg_reader` with formatting options and opens the file named `/tmp/log1` for reading:

```
msg_reader -n /tmp/log1 &
```

The following command prints messages to `stdout` with formatting:

```
msg_reader -n pathname &
```

Use the following command to specify your own parameters:

```
msg_reader [-e] [-n] pathname &
```

# D MIB Reference

A Management Information Base (MIB) describes the attributes of the 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 be formatted in conformity with the SNMP standards. Two MIB are files provided with Log Central that contain information for managing Log Central. These files fully conform to the SNMP standard and are ready for loading into your SNMP manager.

This appendix provides a detailed description of the MIBs that are supported by the SNMP agents shipped with Log Central. The following are included:

- ◆ Log Central Traps MIB
- ◆ Process Monitor MIB

# Log Central Traps MIB

The Log Central Traps (beaTrap) MIB, found in the file `bea_lc_trap.asn1`, contains Log Central attributes that are used as variables in the traps.

Each SNMP trap has a header and body (called variable bindings). The objects in this MIB specify the objects (and their values) that are sent in the variable binding of SNMP traps generated by Log Central. So when a trap is received by an SNMP Manager, it has more information about the condition that generated the trap.

The SNMP Traps MIB contains only the beaTrap group.

Variable Name	Object ID
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

Syntax	INTEGER { normal(78), verbose(86), debug(68), special(83) }
Access	read-only
Description	This object contains the log level of the message for which the SNMP trap was generated.

## beaTrapLcTimestamp

Syntax	OCTET STRING
Access	read-only
Description	This object contains the generation time of the message for which the SNMP trap was generated.

## beaTrapLcSubsys

Syntax	OCTET STRING
Access	read-only
Description	This object contains the originating subsystem name of the message for which the SNMP trap was generated.

## beaTrapLcMid

Syntax	INTEGER
Access	read-only
Description	This object contains the message ID of the message for which the SNMP trap was generated.

### beaTrapLcHost

Syntax	OCTET STRING
Access	read-only
Description	This object contains the originating host name of the message for which the SNMP trap was generated.

### beaTrapLcPid

Syntax	INTEGER
Access	read-only
Description	This object contains the process ID of the message for which the SNMP trap was generated.

### beaTrapLcUid

Syntax	OCTET STRING
Access	read-only
Description	This object contains the user ID responsible for the message for which the SNMP trap was generated.

### beaTrapLcFunction

Syntax	OCTET STRING
Access	read-only
Description	This object contains the function name that generated the message for which the SNMP trap was generated.

## beaTrapLcTxKey

Syntax	OCTET STRING
Access	read-only
Description	This object contains the transaction key (if the originating process was in the context of a transaction) of the message for which the SNMP trap was generated.

## beaTrapLcVersion

Syntax	INTEGER
Access	read-only
Description	This object is currently unused.

## beaTrapLcSeverity

Syntax	INTEGER { informational(1), warning(2), minor(3), major(4), critical(5) unknown(100) }
Access	read-only
Description	This object contains the severity of the message for which the SNMP trap was generated. This is known for traps generated by the message processor on the central host only.

## beaTrapLcMessageBody

Syntax	OCTET STRING
Access	read-only
Description	This object contains the body of the message for which the SNMP trap was generated.

# Process Monitor MIB

The Process Monitor (beaPm) MIB, found in the BEA Manager MIB file (bea.asn1), defines objects that support the Log Central Process Monitor. This MIB group is supported by the pm\_snmpd SMUX subagent provided with the Agent Integrator.

Variable Name	Object ID
beaPmProcsEnvVar	.1.3.6.1.4.1.140.4.1
beaPmMonitorPid	.1.3.6.1.4.1.140.4.2
beaPmMonitorTimer	.1.3.6.1.4.1.140.4.3
beaPmMonitorLastWakeup	.1.3.6.1.4.1.140.4.4
beaPmMaxProcRestarts	.1.3.6.1.4.1.140.4.8
beaPmMaxProcRestartsIntvl	.1.3.6.1.4.1.140.4.9
beaPmProcTable	.1.3.6.1.4.1.140.4.11

## beaPmProcsEnvVar

Syntax	DisplayString
Access	read-only
Description	This object contains the environment variable specifying the interprocess communication (IPC) key used for the BEA Manager proc_monitor.

## beaPmMonitorPid

Syntax	Integer
Access	read-only
Description	This object contains the process ID of proc_monitor.

## beaPmMonitorTimer

Syntax	Integer
Access	read-write
Description	This object contains the wakeup interval of <code>proc_monitor</code> .

## beaPmMonitorLastWakeup

Syntax	Integer
Access	read-only
Description	This object contains the timestamp for the last wakeup of <code>proc_monitor</code> , measured from some epoch. Because <code>proc_monitor</code> monitors all other processes and is a single point of failure, this object is created to monitor the <code>proc_monitor</code> process from the SNMP manager and restart it if necessary.

## beaPmMaxProcRestarts

Syntax	Integer
Access	read-write
Description	This object contains the maximum number of times a process may be restarted by <code>proc_monitor</code> within the time specified by the <code>beaPmMaxProcRestartsIntvl</code> object before <code>proc_monitor</code> stops restarting it and marks the process with a status of crashed.

## beaPmMaxProcRestartsIntvl

Syntax	Integer
Access	read-write
Description	This object specifies the time interval during which a process can be restarted by <code>proc_monitor</code> before it is no longer restarted.

# beaPmProcTable

This object contains row entries. Each entry (row) in the beaPmProcTable is a sequence of the following columnar objects. Each entry represents attributes of processes registered with the BEA Manager `proc_monitor`.

Variable Name	Object ID
beaPmTblEntityName	.1.3.6.1.4.1.140.4.11.1.1
beaPmTblProcId	.1.3.6.1.4.1.140.4.11.1.2
beaPmTblStatus	.1.3.6.1.4.1.140.4.11.1.3
beaPmTblFirstRegTime	.1.3.6.1.4.1.140.4.11.1.4
beaPmTblLastRegTime	.1.3.6.1.4.1.140.4.11.1.5
beaPmTblRestartEnabled	.1.3.6.1.4.1.140.4.11.1.6
beaPmTblRestartCmd	.1.3.6.1.4.1.140.4.11.1.7
beaPmTblNumRestarts	.1.3.6.1.4.1.140.4.11.1.8
beaPmTblRefRestartTime	.1.3.6.1.4.1.140.4.11.1.9
beaPmTblNumRestartsFromRef	.1.3.6.1.4.1.140.4.11.1.10
beaPmTblFirstRestartTime	.1.3.6.1.4.1.140.4.11.1.11
beaPmTblLastRestartTime	.1.3.6.1.4.1.140.4.11.1.12
beaPmTblUid	.1.3.6.1.4.1.140.4.11.1.13
beaPmTblGid	.1.3.6.1.4.1.140.4.11.1.14
beaPmTblEuid	.1.3.6.1.4.1.140.4.11.1.15
beaPmTblEgid	.1.3.6.1.4.1.140.4.11.1.16
beaPmTblSwName	.1.3.6.1.4.1.140.4.11.1.17
beaPmTblSwVersion	.1.3.6.1.4.1.140.4.11.1.18
beaPmTblSwDate	.1.3.6.1.4.1.140.4.11.1.19
beaPmTblSwTime	.1.3.6.1.4.1.140.4.11.1.20

## beaPmTblEntityName

Syntax	DisplayString
Access	read-only
Description	This object contains a name that uniquely identifies an entry in the table.

## beaPmTblProclId

Syntax	Integer
Access	read-only
Description	This object contains the process ID of the registered process.

## beaPmTblStatus

Syntax	Integer {uninitialized (1), running (2), restarted (3), dead (4), unregistered (5), crashed (6) }
Access	read-only
Description	This object contains the current status of the registered process. Processes are marked <code>unregistered</code> if they have died and are not to be restarted. A process is marked <code>dead</code> if <code>proc_monitor</code> detected that it is no longer running. A process is marked <code>restarted</code> if <code>proc_monitor</code> restarted it. A process is marked <code>crashed</code> if it was restarted too many times in a specified interval of time.

## beaPmTblFirstRegTime

Syntax	DisplayString
Access	read-only
Description	This object contains the time when the process first registered.

### beaPmTblLastRegTime

Syntax	DisplayString
Access	read-only
Description	This object contains the time when the process last registered.

### beaPmTblRestartEnabled

Syntax	Integer {no (1), yes (2), nomore (3) }
Access	read-only
Description	This object contains a flag to indicate if a process can be restarted by <code>proc_monitor</code> .

### beaPmTblRestartCmd

Syntax	DisplayString
Access	read-only
Description	This object contains the command to be executed by <code>proc_monitor</code> when restarting a process if this flag is <code>yes</code> .

### beaPmTblNumRestarts

Syntax	Integer
Access	read-only
Description	This object contains the number of times this process has been restarted.

### beaPmTblRefRestartTime

Syntax	DisplayString
Access	read-only
Description	This object contains the reference timestamp from which <code>proc_monitor</code> must determine if a process should be restarted or marked as crashed.



## beaPmTblNumRestartsFromRef

Syntax	Integer
Access	read-only
Description	This object contains the number of restarts that occurred since the <code>beaPmTblRefRestartTime</code> timestamp.

## beaPmTblFirstRestartTime

Syntax	DisplayString
Access	read-only
Description	This object contains the timestamp of the first restart of the process.

## beaPmTblLastRestartTime

Syntax	DisplayString
Access	read-only
Description	This object contains the timestamp of the last restart of the process.

## beaPmTblUid

Syntax	Integer
Access	read-only
Description	This object contains the process user ID. This object is saved when the process registers so that the process can be set to the same user ID if it is restarted.

## beaPmTblGid

Syntax	Integer
Access	read-only
Description	This object contains the group ID of the owner of the process. This object is saved when the process registers so that the group ID can be set to the same value if the process is restarted.

### beaPmTblEuid

Syntax	Integer
Access	read-only
Description	This object contains the effective user ID. This object is saved when the process registers so that the process can be set to the same value if it is restarted by <code>proc_monitor</code> .

### beaPmTblEgid

Syntax	Integer
Access	read-only
Description	This object contains the effective group ID. This object is saved when the process registers so that the process can be set to the same value if it is restarted by <code>proc_monitor</code> .

### beaPmTblSwName

Syntax	DisplayString
Access	read-only
Description	This object contains the software subsystem name of the process. This object is used for configuration management to ensure that a process is an execution of the correct version of the software.

### beaPmTblSwVersion

Syntax	DisplayString
Access	read-only
Description	This object contains the software version of the process. This object is used for configuration management to ensure that a process is an execution of the correct version of the software.

**beaPmTblSwDate**

Syntax	DisplayString
Access	read-only
Description	This object contains the software creation date of the process. This object is used for configuration management to ensure that a process is an execution of the correct version of the software.

**beaPmTblSwTime**

Syntax	DisplayString
Access	read-only
Description	This object contains the software creation time of the process. This object is used for configuration management to ensure that a process is an execution of the correct version of the software.



# E Database Schema

A database administrator needs to know the contents of the Log Central database table definitions to help determine which tables to back up and when to back them up.

This appendix summarizes all the database table definitions used by Log Central. It includes the following tables:

- ◆ Schema Tables
- ◆ Log Message Definitions
- ◆ Logging Levels
- ◆ Message Severities
- ◆ Subsystem Definitions
- ◆ Trap Classes
- ◆ Logged Message Data
- ◆ User Data

# Schema Tables

Table E-1 lists the schema tables.

**Table E-1   Table Summary**

Table Name	Description
IL_MSG	Contains data pertaining to log message definitions.
IL_REP_MODE	Contains data pertaining to the log reporting modes.
IL_SEV	Contains data pertaining to the message severities.
IL_SUBS	Contains data pertaining to subsystem definitions.
IL_TRAP_CLASS	Contains data pertaining to SNMP trap classes.
IL_SM_LOG	Contains data pertaining to logged messages.
IL_USER	Contains data pertaining to users.

# Log Message Definitions

Table E-2 IL\_MSG

Field Name	Null?	Type	Size	Description
LOG_SUBS_NAME	Not null	CHAR	8	The name of the subsystem
LOG_MSG_ID	Not null	NUMBER	5	The message ID number
LOG_SDESC	Not null	CHAR	40	A short description of the log message
LOG_MSG_SEVERITY	Not null	NUMBER	1	The severity level.
LOG_REP_MODE	Not null	CHAR	1	The log reporting mode.
LOG_MSG_VERSION	Not null	NUMBER	3	The message version number.
LOG_MSG_MNEMONIC		CHAR	40	The name correlating a message to the software module in which it is defined.
LOG_MSG_PARSE_FNC		CHAR	40	The program to be executed when a live message arrives that maps with this message definition.
LOG_MSG_TRAP_ID		NUMBER	6	The ID number for the SNMP trap.
LOG_MSG_TRAP_ENABLED		NUMBER	1	The SNMP trap flag.
LOG_MSG_AUTO_ACK	Not null	NUMBER	1	Whether the message is auto-acknowledged.
LOG_MSG_DESC_REC	Not null	LONG		Description and recommendation in the following format: <ul style="list-style-type: none"><li>◆ 5 digits representing the length of the description</li><li>◆ the description itself</li><li>◆ the recommendation, occupying the remainder of the field</li></ul>

# Logging Levels

Table E-3 IL\_REP\_MODE

Field Name	Null?	Type	Size	Description
LOG_REP_MODE_ID	Not null	CHAR	1	The character that identifies the logging level. Valid values are N, V, D, and S.
LOG_REP_MODE_DESC	Not null	CHAR	7	The string that describes the logging level. Valid values are: Normal, Verbose, Debug, and Special.

# Message Severities

Table E-4 IL\_SEV

Field Name	Null?	Type	Size	Description
LOG_SEV_ID	Not null	NUMBER	1	A number that identifies the severity level. Valid values are: 1, 2, 3, 4, 5.
LOG_SEV_DESC	Not null	CHAR	13	The string that describes the severity level. Valid values are: Informational, Warning, Minor, Major, Critical.
LOG_SEV_TRAP_ID	Not null	NUMBER	1	The ID number for the SNMP trap, based on message severity. Valid values are any 6-digit number.
LOG_SEV_TRAP_ENABLED	Not null	NUMBER	1	The SNMP trap flag, based on message severity. Valid values are 0 and 1.



# Subsystem Definitions

**Table E-5 IL\_SUBS**

Field Name	Null?	Type	Size	Description
LOG_SUBS_NAME	Not null	CHAR	8	A string that identifies the subsystem name.
LOG_SUBS_DESC		CHAR	40	A string that describes the subsystem name.

## Trap Classes

**Table E-6 IL\_TRAP\_CLASS**

Field Name	Null?	Type	Size	Description
LOG_TRAP_CLASS_NAME	Not null	CHAR	16	The name of the SNMP trap class. Valid values are: <i>message</i> and <i>severity</i> .
LOG_TRAP_CLASS_ENABLED	Not null	NUMBER	1	A flag to enable or disable SNMP traps in the specified class. Valid values are 0 and 1.

# Logged Message Data

Table E-7 IL\_SM\_LOG TABLE

Field Name	Null?	Type	Size	Description
MSG_KEY_TS	Not null	DATE		A time stamp indicating when the message was logged.
MSG_KEY_CTR	Not null	NUMBER	3	A counter for multiple messages received in the same second.
MSG_LOG_ID		CHAR	3	The log ID string.
MSG_REP_MODE	Not null	CHAR	1	The logging level.
MSG_SEVERITY	Not null	NUMBER	1	The message severity.
MSG_SS_NAME	Not null	CHAR	8	The subsystem name.
MSG_ID	Not null	NUMBER	5	The message ID number.
MSG_HOST_NAME	Not null	CHAR	20	The name of the sending host.
MSG_PID	Not null	NUMBER	5	The process ID.
MSG_UID	Not null	CHAR	8	The user ID.
MSG_FCT_NAME		CHAR	40	The name of the function.
MSG_TRAN_KEY		CHAR	21	The transaction ID.
MSG_VERSION	Not null	NUMBER	3	The message body version number. This is a reserved field.
MSG_TEXT		LONG	2000	The message body.

# User Data

Table E-8 IL\_USER TABLE

Field Name	Null?	Type	Size	Description
USERNAME	Not null	CHAR	10	User name.
PASSWORD	Not null	CHAR	10	Password.
FIELDS		CHAR	100	Displayed fields. Stores list of fields displayed in the Message Definition Browser.
ORDERS		CHAR	100	Field order. Stores the order in which the display fields will be seen.



# F Initialization File

## Initialization File

The Log Central database management and the Log Central user interface get their initialization information from the `installdir/etc/msg_processor.ini` file.

To fine-tune system performance, you may wish to modify the values of some of the parameters of the initialization file. You might, for example, want to change the default values of sleep times and read intervals. Also, on some systems, you might need to specify Oracle thin drivers in the database URL.

The following table lists all the parameters that can be set within `msg_processor.ini`.

Properties Name	Description	Valid Values	Interested Programs
LC.URL	URL for database Examples MS SQL Server or Oracle Type-1 driver <code>jdbc:odbc:mngrdb</code> Oracle8 Type-2 driver <code>jdbc:oracle:oci8:@lcdbserv</code> Oracle Type-4 driver <code>jdbc:oracle:thin:@amazon:1521:amazonProd</code>		<code>msg_processor</code> , <code>lc_create_schema</code> , <code>lc_drop_schema</code> , <code>msgdef_export</code> , <code>msgdef_import</code> , <code>msgdef_delete</code> , <code>subsystem_create</code> , <code>subsystem_delete</code> , <code>lc_user_commands</code>

Properties Name	Description	Valid Values	Interested Programs
LC.driver	JDBC Driver Name Examples MS SQL Server or Oracle Type-1 driver <code>sun.jdbc.odbc.JdbcOdbcDriver</code> Oracle Type-2 or Type-4 driver <code>oracle.jdbc.driver.OracleDriver</code>		<code>msg_processor</code> , <code>lc_create_schema</code> , <code>lc_drop_schema</code> , <code>msgdef_export</code> , <code>msgdef_import</code> , <code>msgdef_delete</code> , <code>subsystem_create</code> , <code>subsystem_delete</code> , <code>lc_user_commands</code>
LC.userName	Database login name Default/Example <code>scott</code>		<code>msg_processor</code> , <code>lc_create_schema</code> , <code>lc_drop_schema</code> , <code>msgdef_export</code> , <code>msgdef_import</code> , <code>msgdef_delete</code> , <code>subsystem_create</code> , <code>subsystem_delete</code> , <code>lc_user_commands</code>
LC.password	Database password Default/Example <code>tiger</code>		<code>msg_processor</code> , <code>lc_create_schema</code> , <code>lc_drop_schema</code> , <code>msgdef_export</code> , <code>msgdef_import</code> , <code>msgdef_delete</code> , <code>subsystem_create</code> , <code>subsystem_delete</code> , <code>lc_user_commands</code>
LC.DBVendor	Database vendor name Default/Example <code>ORACLE</code>	<code>ORACLE</code> <code>, MSSQL</code>	<code>msg_processor</code> , <code>lc_create_schema</code> , <code>lc_drop_schema</code> , <code>msgdef_export</code> , <code>msgdef_import</code> , <code>msgdef_delete</code> , <code>subsystem_create</code> , <code>subsystem_delete</code> , <code>lc_user_commands</code>

Properties Name	Description	Valid Values	Interested Programs
LC.parseFunction	Process parse function or not Default true	true, false	msg_processor
LC.loadDB	Database is available or not Default true	true, false	msg_processor
LC.maxOpenTries	Maximum number of attempts to open a file Default 10000	numeric value	msg_processor
LC.openSleepTime	Sleep time in milliseconds if file is not available for open Default 2000	numeric value	msg_processor
LC.readSleepTime	Sleep time in milliseconds between two reads if no new records Default 2000	numeric value	msg_processor
LC.readDelayTime	Delay time in milliseconds between two reads; should be 0 Default 0	numeric value	msg_processor
LC.realTimeSleep Time	Sleep time in milliseconds between two reads from shared memory if no new records is available Default 1000		msg_processor

Properties Name	Description	Valid Values	Interested Programs
LC.ILLog	Path for error/debug logs created by IL system itself Default/Example LCLog	valid file names	msg_processor , lc_create_schema , lc_drop_schema , msgdef_export , msgdef_import , msgdef_delete , subsystem_create , subsystem_delete , lc_user_commands
LC.Server.Parameters .MaxConnections	Maximum simultaneous connection allowed; 0 means no limit Default 100	numeric	msg_processor
LC.Server.dbPort	Port for msg_processor Default 7001		msg_processor , lc_launch



# G Predefined Log Mapping

Predefined mappings are available for integrating the following log resources into the Log Central system with Log Monitor:

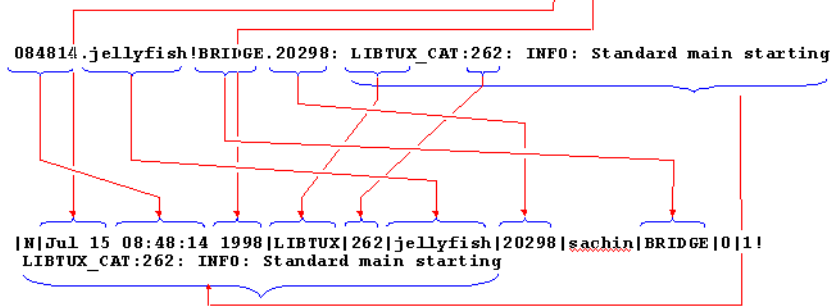
- ◆ BEA TUXEDO Message Mapping
- ◆ Windows NT Event Log
- ◆ Oracle Alert Log

For more information about the storage format of Log Central messages, refer to Chapter 4, “Integrating Logs into Log Central.”

# BEA TUXEDO Message Mapping

The following graphic shows the predefined mapping of a typical BEA TUXEDO ULOG message to a Log Central message.

Typical TUXEDO ULOG message from file ULOG.071598

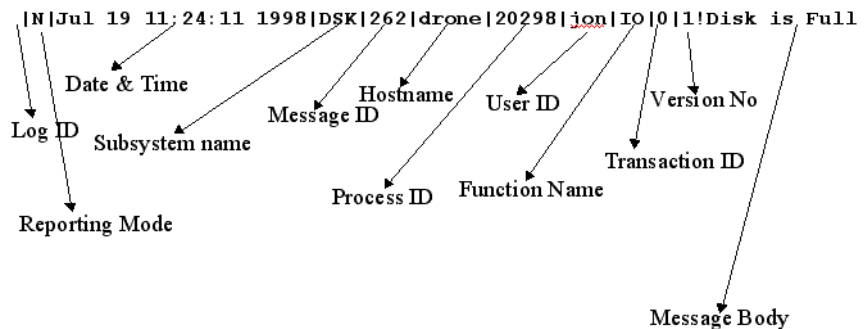


The Log Monitor command to perform this mapping follows:

```
log_monitor -P TUXEDO -i ULOG.071598
```

## Log Central Message Fields

The following graphic shows the fields of a Log Central message.



# Windows NT Event Log

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

Log Central Field	Value
Date	NT Event date + Time
Body	NT Event description
Message ID	NT Event ID
Subsystem	NT Event Source
Host	NT Event Computer
Process ID	PID of log_monitor
Function	None
Transaction ID	0
User ID	NT Event User
Version	1 (If message body is greater than 2000 characters, then multiple messages are sent, incrementing the version number.)

The Log Monitor command to perform this mapping follows:

```
log_monitor -P NTEVENT
```

# Oracle Alert Log

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

Log Central Field	Value
Date	Alert log date
Body	Multiline message in Alert log
Message ID	999
Subsystem	ORACLE
Host	Host where the <code>log_monitor</code> process is running
Process ID	PID of <code>log_monitor</code>
Function	None
Transaction ID	0
User ID	User name of the <code>log_monitor</code> process
Version	1 (If message body is greater than 2000 characters, then multiple messages are sent, incrementing the version number.)

The Log Monitor command to perform this mapping follows:

```
log_monitor -P ORACLE -i alert_log_file
```

---

# Glossary

## **Abstract Syntax Notation One (ASN.1)**

A formal notation used to define data types and encode data values. A language that describes the data structures that make up an abstract syntax. ITU-T (formerly CCITT) specification X.409 is equivalent to ASN.1. ASN.1 is used to define a management information base (MIB). ASN.1 is a common requirement of the SNMP and CMIP network management protocols.

## **activity message**

A log message that primarily provides information about events that occur in the system rather than reporting error conditions.

## **AFS**

Attributed File System

## **agent**

A component of network management that exchanges data about *managed resources* with a network manager. Agents provide a software interface to managed resources, and gather data about them at a manager's request.

## **agent/manager model**

A model where a manager communicates with many distributed *agents* via a network or system management protocol.

## **alarm**

A message reporting that a managed object is in an abnormal state (i.e., a managed object has passed a pre-defined threshold).

## **API**

See *application programming interface*.

## **application programming interface (API)**

A set of calling conventions that define how to invoke a service.

---

**architecture**

The structure and interrelationship of components in a system or in an environment.

**ASN.1**

A standard notation used to define a management information base (MIB). See *Abstract Syntax Notation One (ASN.1)*.

**BEA TUXEDO System**

A robust middleware engine for developing and deploying business-critical client/server applications. It handles distributed transaction processing, application messaging, and the full complement of services necessary to build and run enterprise-wide applications.

**client**

A process that interacts with users and submits requests to a server.

**client/server**

1) A model used in distributed systems where one host acts as a system server, and the other host acts as a client. 2) A distribution model in which there are two types of applications: client applications that request that tasks be performed, and server applications that perform those tasks. 3) A programming model in which application programs are structured as clients or servers. A client program is an application program that requests services to be performed. A server program is an entity that dispatches service routines to satisfy requests from client programs. A service routine is an application program module that performs one or more specific functions on behalf of client programs.

**CMIP**

See *Common Management Information Protocol (CMIP)*.

**Common Management Interface Protocol (CMIP)**

A protocol for network management defined by *ISO* standards. The network management protocol for OSI networks.

**configuration management**

Monitoring the state of an application process (i.e., whether a process is running, stopped, or restarted), and controlling a process' software revision level (i.e., providing the software's current version).

---

**database**

A collection of one or more tables or files under the control of a database management system.

**database management system (DBMS)**

A program or set of programs that let users structure and manipulate the data in the tables of a *database*. A DBMS ensures privacy, recovery, and integrity of data in a multi-user environment.

**DBMS**

See *database management system*.

**EGID**

Effective group identification number.

**enterprise-wide environment**

This term refers to all components in a distributed environment, both local and remote.

**escalation procedure**

An alternative method for logging a log message when the main logging option fails.

**error message**

A log message that indicates that a process detected an error condition which must be recorded in a log.

**EUID**

Effective user identification number.

**event notification**

A means for application processes to communicate with each other about important occurrences in the system (i.e., they send event notifications to each other).

**event management**

A process sending an *event notification* to one or more other processes. Event management is primarily used for setting or modifying configuration parameters in running processes (e.g., modifying their logging options or forcing them to re-read their configuration files).

---

**field**

1) A way of addressing a single item of data in a *database* table. A field is the data that exists at a column/row location in a database table. 2) A field is also an area of a window where data displays.

**GID**

Group identification number.

**graphical user interface (GUI)**

A high-level interface that uses windows and menus with graphic symbols instead of typed system commands to provide an interactive environment for a user.

**GUI**

See *graphical user interface*.

**host**

A computer running application software. A host may be at your site (i.e., local) or at another site (i.e., remote).

**identification string**

Portions of a file that get expanded by BEA Manager utilities to contain file and system identification information. If compiled, these strings are placed into object file functions, where their information is then available. These are also called `ident` strings.

**International Organization for Standardization (ISO)**

An organization that produces standards.

**ISO**

See *International Organization for Standardization*.

**managed object**

1) A software entity, defined in a *Management Information Base* (MIB), that represents a managed resource (such as a process, a piece of hardware, or a system performance attribute), and is controlled by a management device.

2) In *Simple Network Management Protocol*, a managed object represents a manageable attribute of a managed resource, such as the average transit time of transactions through an application or the number of interface cards in a host computer.



---

**managed resource**

The physical resource whose attributes are represented by *managed objects* in a *Management Information Base*. A managed resource can be a software entity such as an application or queue, or a hardware device, such as an interface card or hub.

**management framework**

A system that provides a unified view of hardware and software resources on distributed systems and enterprise-wide networks to aid network administrators to manage and control these resources.

**management information base (MIB)**

- 1) A virtual storage database that uses *ASN.1* notation. The MIB contains each object that the system management software monitors and controls. Each has a unique object name and a unique *object identifier*.
- 2) A *BEA TUXEDO* System component that provides a complete definition of the object classes and their attributes that together comprise the *BEA TUXEDO* System. The total *BEA TUXEDO* System Management Information Base is organized into a generic MIB and component-specific MIBs for each major component. Configuration and administration of the *BEA TUXEDO* System can be done programmatically.

**manager (system manager)**

A component of network management that requests data from an agent, and takes actions based on that data.

**message (log message)**

A means for sending data and values across applications. Messages represent statistical or status information about application processes, and consist of a header, containing message ID data; and a body, containing user-defined information.

**message definition block**

The total body of data that comprises a message definition, such as the command name, the subsystem name, and the internal and external recommendations, collectively.

**MIB**

See *management information base (MIB)*.

**MIB group**

A group of objects, represented by the name or OID of an object in the OID (or

---

registration) tree, which contains a collection of managed objects. A MIB group may contain other MIB groups, or it may contain scalar or tabular objects.

**node application**

A piece of application software that you can invoke on a host.

**MOPS**

Management operations per second.

**object identifier (OID)**

OIDs are unique designators of attributes of managed resources and are used in both SNMP and CMIP network management. An OID is a unique number that defines an object in the MIB. The numbers that comprise the OID describe a path to the object through a tree hierarchy, which is often called an OID tree or a registration tree. When the SNMP agent wishes to access a specific object, it traverses the OID tree in the MIB file to find the object.

**OID**

See *object identifier*.

**OLTP**

See *online transaction processing*.

**online transaction processing (OLTP)**

The execution of units of work in a performance-critical environment that appears to the user as immediate; real-time; usually having internal recoverability, history-keeping and consistency-assurance features.

**Open Systems Interconnection (OSI)**

A consortium that facilitates communications between different types of computer systems.

**OSI**

See *Open Systems Interconnection*.

**PID**

See *process ID (PID)*.

**polling**

---

An activity where a manager interrogates an agent at a periodic interval. The agents report the values of the specified managed objects. Checking at periodic intervals to determine if a managed object value has crossed a specified threshold.

**private MIB**

A MIB that is defined under the private MIB directory. Private MIBs contain objects that are specific to a company's software needs. The identification number of the BEA private MIB is 1.3.6.1.4.1.140.

**process events**

Event notifications that cause a process to execute a specific task when they are delivered.

**process ID (PID)**

A unique number that identifies a process.

**process management**

Managing the state of an application process (i.e., whether a process is running, stopped, recovered, or restarted) and its software version.

**register a process**

A term used to describe how a process provides information about itself in shared memory for event management and/or process management. Once a process registers, process management software can track it.

**requestor**

A process that receives messages from clients, converts these messages to a common internal form, determines the appropriate server or servers for the transaction request, and forwards the request to a server.

**RCS**

Revision control system.

**RDBMS**

Relational database management system.

**server**

A process that provides services to requesting processes. Each server includes one or more services, each of which performs specific transaction processing functions.

---

**severity**

The *severity* of a message or event notification is a rating used to represent the importance or impact of the event being reported. For example, a message that indicates high usage of network bandwidth reports a less severe event than a message telling you that a server has crashed.

**shared memory manager (SMGR)**

The Shared Memory Manager subsystem, which provides mechanisms that enable you to manage a set of shared memory segments. You can use SMGR as a basic building block for any application that require processes to communicate through shared memory.

**Simple Network Management Protocol (SNMP)**

A de-facto standard network management protocol developed by the Internet community.

**SMGR**

See *shared memory manager*.

**SNMP**

See *Simple Network Management Protocol (SNMP)*.

**SNMP agent**

An agent using the *SNMP* protocol to exchange data with the system manager.

**SNMP mask**

A means for hiding selected SNMP traps, letting you cause alarms only in certain instances.

**standard MIB**

A MIB developed as a standard by the Internet community. Examples are MIB I and MIB II.

**TCP/IP**

See *Transmission Control Protocol/Internet Protocol*.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

A network protocol that is part of the Internet suite of protocols.

**Timer Events**

---

Events that signal the passing of time to a process. This tells the process that it is time to execute a specific task.

**Token**

An individual element in the message definition block, such as the command or the subsystem name.

**trap**

An *SNMP* data packet containing information about an error that occurred with a managed object. Traps are unsolicited event notifications, that is, notifications generated by an agent on its own initiative.

**UDP**

See *User Datagram Protocol*.

**UID**

User identification number.

**User Datagram Protocol (UDP)**

The TCP/IP datagram transport layer protocol.

**window**

An area of a user's screen in a Graphical User Interface system. A window is a mechanism used by applications for interacting with a user.



---

# Index

## Symbols

#

in configuration file 6-2

## A

acknowledge messages, Message Browser  
9-9

add a new message definition, Message  
Definition Editor 9-28

advanced trap configuration 6-13

agent filter in LC\_GLOBAL entry 6-6

agent filters 6-9

Agent Integrator polling rules 7-5

AUTO\_ACKNOWLEDGE, in message  
definition 5-5

automatic acknowledgment, attribute in  
message definition A-8

## B

backup Central Collector 6-3

BACKUP\_HOST, LC\_GLOBAL keyword  
6-5

BACKUP\_LOGPREFIX, LC\_GLOBAL  
keyword 6-5

Basic Trap Configuration

procedures for using 9-42

trap generation

by message definition 9-45

by message definition and severity  
9-46

by severity 9-43

BEA TUXEDO message mapping,  
predefined G-2

BEA\_LC\_CONF\_SERVICE B-3

BEA\_LC\_IPCKEY B-1

BEA\_SM\_BEAMGR\_CONF B-4

beaEmMaxProcRestarts D-7

beaEmMaxProcRestartsIntvl D-7

beaEmMonitorLastWakeup D-7

beaEmMonitorPid D-6

beaEmMonitorTimer D-7

beaEmProcEnvVar D-6

beaEmProcV2Table D-8

beaEmTblEgid D-12

beaEmTblEntityName D-9

beaEmTblEuid D-12

beaEmTblFirstRegTime D-9

beaEmTblFirstRestartTime D-11

beaEmTblGid D-11

beaEmTblLastRegTime D-10

beaEmTblLastRestartTime D-11

beaEmTblNumRestarts D-10

beaEmTblNumRestartsFromRef D-11

beaEmTblProcId D-9

beaEmTblRefRestartTime D-10

beaEmTblRestartCmd D-10

beaEmTblRestartEnabled D-10

beaEmTblStatus D-9

beaEmTblSwDate D-13

---

beaEmTblSwName D-12  
beaEmTblSwTime D-13  
beaEmTblSwVersion D-12  
beaEmTblUid D-11  
beaSystemDescr, in HP OpenView 7-5  
beaTrapLcHost D-4  
beaTrapLcLogLevel D-3  
beaTrapLcMessageBody D-5  
beaTrapLcMid D-3  
beaTrapLcPid D-4  
beaTrapLcSeverity D-5  
beaTrapLcSubsys D-3  
beaTrapLcTimestamp D-3  
beaTrapLcTxKey D-5  
beaTrapLcUid D-4  
beaTrapLcVersion D-5  
body, of message, message attribute, at agent  
A-5

## C

Central Collector  
    backup, specifying 6-3  
Central Collector SNMP trap generation 7-3  
Central Collector, functions of 1-9  
CENTRAL\_HOST, LC\_GLOBAL keyword  
    6-5  
colors, change  
    Message Browser 9-17  
    Message Definition Editor 9-34  
COMMAND, DEFINE\_FILTER action 6-14  
comment  
    in configuration file 6-2  
configuration file  
    comments 6-2  
    comments in 6-2  
    description 4-2  
    number of mappings 4-2  
    pound sign 6-2  
configuration file format symbols, Log  
    Monitor 4-5

configuration file, with Log Monitor  
    command line mappings 8-9  
    description 8-8  
    number of mappings 8-8  
configuration management, with Event  
    Manager  
    MIB objects useful for D-12  
configuration, host and filter 6-1  
current log file, reading C-3

## D

data collection agent, SNMP trap generation  
    7-3  
data collection system  
    starting 8-2  
    stopping 8-12  
database records  
    manual deletion, Storage Maintenance  
        9-53  
    scheduling deletion, Storage  
        Maintenance 9-52  
    scheduling processing and deletion of,  
        Storage Maintenance 9-56  
    scheduling processing, Storage  
        Maintenance 9-54  
database schema, reference E-1  
database, Log Central, access via message  
    browser 1-9  
date and time, message attribute, at agent A-3  
date format, Log Monitor configuration file  
    4-11  
date format, Log Monitor configuration file,  
    %f 4-15  
default configuration, host and filter 6-2  
DEFINE\_FILTER  
    action keywords 6-14  
    dropping a message 6-12  
    example 6-14  
    executing a command 6-13  
    keywords in 6-11



---

- relational conditions 6-12
- SNMP trap notification 6-13
- syntax 6-9
- delete message definitions, Message Definition Editor 9-30
- delete messages, Message Browser 9-8
- deletion of database records
  - manual, Storage Maintenance 9-53
  - scheduling, Storage Maintenance 9-52
- deletion of intermediate files, scheduling, Storage Maintenance 9-57
- description, attribute in message definition A-7
- DESCRIPTION, in message definition 5-4
- Displayed/Matched field
  - Message Browser 9-5
  - Message Definition Editor 9-26
- dropping a message, DEFINE\_FILTER 6-12

## E

- enterprise identifiers 7-6
- environment variables
  - BEA\_LC\_CONF\_SERVICE B-3
  - BEA\_LC\_IPCKEY B-1
  - BEA\_SM\_BEAMGR\_CONF B-4
  - reference B-1
- environment, JRE 2-2
- ESCALATION\_DIR, LC\_GLOBAL keyword 6-5
- Event Manager
  - MIB objects for D-6, D-8
- example, DEFINE\_FILTER 6-14
- EXECUTE\_ON\_UPLOAD, in message definition 5-4
- executing a command, DEFINE\_FILTER 6-13

## F

- field lengths, Log Monitor configuration file 4-10
- filter message definitions, Message Definition Editor 9-32
- filter messages, Message Browser 9-10
- FILTER, LC\_GLOBAL keyword 6-5
- FILTER, MANAGED\_NODE entry 6-8
- filtering system log, Log Monitor configuration file 4-12
- filtering, at Central Collector 1-10
- filters
  - assigning to agents 6-6
  - assigning to managed node 6-8
- function name, message attribute, at agent A-5
- FUNCTION, DEFINE\_FILTER 6-11

## G

- generate reports
  - Message Browser 9-19
  - Message Definition Editor 9-37
- generating test messages C-1
- global filters
  - nondefault, configuration file 6-4
  - turning off on a managed node 6-7
- GLOBAL\_FILTER, MANAGED\_NODE entry 6-8

## H

- historical database queries 9-7
- historical queries, Message Browser 9-7
- host and filter configuration 6-1
- host name usage, configuration file 6-3
- host name, message attribute, at agent A-4
- HOST, DEFINE\_FILTER 6-11
- HP OpenView, enterprise entry 7-5

---

## I

- IL\_MSG E-3
- IL\_REP\_MODE E-4
- IL\_SEV E-4
- IL\_SM\_LOG TABLE E-6
- IL\_SUB E-5
- IL\_TRAP\_CLASS E-5
- IL\_USER TABLE E-7
- INIFILE, LC\_GLOBAL keyword 6-5
- initialization file, reference F-1
- input dates, converting, with Log Monitor
  - configuration file 4-15
- install\_dir, definition 2-6
- intermediate log files, where to locate 6-2
- IPC key value, default, IPCKEY 6-4
- IPCKEY B-1
- IPCKEY, LC\_GLOBAL keyword 6-5

## J

- JRE, adding path to environment 2-2

## L

- Launch Panel
  - appearance 9-2
  - invoking 9-1
- layout, change
  - Message Browser 9-16
  - Message Definition Editor 9-33
- lc\_create\_schema, syntax 3-2
- lc\_drop\_schema, syntax 3-2
- LC\_GLOBAL entry, with agent filter 6-6
- LC\_GLOBAL keywords, table of 6-5
- LC\_GLOBAL, default configuration 6-2
- lc\_talk, default service name for msg\_sender
  - processes 6-4
- LOCAL, DEFINE\_FILTER action 6-14
- Log Central
  - starting and stopping 8-1
  - starting on a central host 8-3

- starting on managed nodes 8-5
- Log Central Console, procedures for using 9-1
- Log Central data collection system
  - starting 8-2
  - stopping 8-12
- Log Central database, getting message
  - definitions into 5-2
- Log Central system information, displaying 8-13
- Log Central Traps MIB, reference D-2
- log ID A-3
- log message definitions, database schema E-3
- Log Monitor
  - configuration file format symbols 4-5
  - configuration file options 4-3
  - using with configuration file, example 4-6
- Log Monitor configuration file
  - converting input dates 4-15
  - date format 4-11
  - field lengths 4-10
  - filtering system log 4-12
  - multiple separator characters 4-7
  - pattern selection 4-8
- Log Monitor, description 1-7
- log reporting modes, database schema E-4
- LOG\_LEVEL, DEFINE\_FILTER 6-11
- logged message data, database schema E-6
- logging level, message attribute, at agent A-3
- LOGPREFIX
  - intermediate log files 6-2
  - LC\_GLOBAL keyword 6-5

## M

- main window
  - Message Definition Editor 9-25
  - Storage Maintenance 9-48

---

- managed nodes
  - assigning filters to 6-8
  - starting Log Central on 8-5
  - turning off global filters 6-7
- MANAGED\_NODE
  - keywords, table 6-8
  - using 6-7
- managing Log Central, SNMP 7-2
- message attribute, at agent
  - body of message A-5
  - date and time A-3
  - function name A-5
  - host name A-4
  - logging level A-3
  - message ID A-4
  - process ID A-5
  - subsystem name A-4
  - transaction ID A-5
  - user ID A-5
- Message Browser
  - acknowledge messages 9-9
  - change colors 9-17
  - change layout 9-16
  - delete messages 9-8
  - filter messages 9-10
  - generate reports 9-19
  - main window 9-4
  - remove acknowledgment from messages 9-10
  - using, procedures for 9-3
  - view message details 9-19
- message components A-2
- Message Definition Editor
  - add a new message definition 9-28
  - change colors 9-34
  - change layout 9-33
  - delete message definitions 9-30
  - filter message definitions 9-32
  - generate reports 9-37
  - main window 9-25
  - modify a message definition 9-30
  - modify subsystem form 9-36
  - procedures for using 9-24
- message definition file
  - creating 5-3
  - loading 5-6
- message definition, attribute in
  - automatic acknowledgment A-8
  - description A-7
  - recommendation A-7
  - severity A-6
  - summary A-7
  - trap enabled A-7
  - trap generation A-7
  - trap ID A-8
- message definition, example 5-5
- message definitions
  - creating and loading 5-1
  - deleting, msgdef\_delete 5-8
  - deleting, subsystem\_delete 5-8
  - description 5-1
  - exporting 5-7
  - filter, Message Definition Editor 9-32
  - getting into Log Central database 5-2
  - modify Message Definition Editor 9-30
- message details, view, Message Browser 9-19
- message format
  - message attribute, at agent A-4
  - reference A-1
- message ID, message attribute, at agent A-4
- Message Processor, starting 8-3
- Message Receiver, starting 8-3
- Message Sender
  - description 1-8
  - starting 8-3
- message severities, database schema E-4
- MESSAGE\_ID, in message definition 5-4
- MIB reference D-1
- modify a message definition, Message Definition Editor 9-30

---

- monitor incoming messages, Message Browser 9-7
- monitoring, role of Central Collector in 1-9
- msg\_processor, starting 8-3
- msg\_processor.ini, parameters F-1
- msg\_reader command
  - description C-3
  - options C-3
  - syntax C-3
- msg\_receiver, starting 8-3
- msg\_sender
  - processes, default service name for 6-4
  - starting 8-3
- msg\_test command
  - options C-1
  - syntax C-1
- MSGBODY, DEFINE\_FILTER 6-11
- msgdef\_delete 5-8
- MSGID, DEFINE\_FILTER 6-11

## O

- object identifier (OID) for Log Central 7-5
- OID for Log Central 7-5
- Oracle event log message mapping,
  - predefined G-4

## P

- pattern selection, Log Monitor configuration file 4-8
- pm\_snmpd, MIB objects supported by D-6
- pound sign in configuration file 6-2
- predefined log mapping G-1
- predefined message mapping
  - BEA TUXEDO G-2
  - Oracle event log G-4
  - Windows NT event log G-3
- proc\_monitor, starting 8-3
- process ID, message attribute, at agent A-5
- Process Monitor, starting 8-3
- process\_dbrec script, Storage Maintenance 9-51

- processing and deletion of database records,
  - scheduling, Storage Maintenance 9-56
- processing database records, scheduling, Storage Maintenance 9-54

## R

- recommendation, attribute in message
  - definition A-7
- RECOMMENDATION, in message
  - definition 5-4
- records processing script, preparing, Storage Maintenance 9-50
- relational conditions, DEFINE\_FILTER 6-12
- REMOTE, DEFINE\_FILTER action 6-14
- remove acknowledgment from messages, Message Browser 9-10
- reports generation
  - Message Browser 9-19
  - Message Definition Editor 9-37

## S

- separator characters with Log Monitor
  - configuration file 4-7
- severity, attribute in message definition A-6
- SEVERITY, in message definition 5-4
- show\_config, displaying Log Central system information 8-13
- SMUX subagent 7-3
- SNMP integration, overview 7-1
- SNMP management, setting up in Log Central 7-4
- SNMP Multiplex Protocol (SMUX) subagent 7-3
- SNMP trap generation, Central Collector 7-3
- SNMP trap generation, data collection agent 7-3
- SNMP trap notification, DEFINE\_FILTER 6-13

---

## SNMP traps

- configuring for data collection agents 6-13

- SNMP, managing Log Central 7-2

- start\_messaging command 8-5

- start\_messaging command, starting Log Central components 8-3

- starting and stopping Log Central 8-1

- starting Log Central

  - on a central host 8-3, 8-4

  - on managed node 8-5

  - on managed nodes 8-5

- stop\_messaging, stopping Log Central data collection system 8-12

## Storage Maintenance

- main window 9-48

- manual deletion of database records 9-53

- preparing records processing script 9-50

- procedures for using 9-47

- process\_dbrec script 9-51

- scheduling deletion of database records 9-52

- scheduling deletion of intermediate files 9-57

- scheduling processing and deletion of database records 9-56

- scheduling processing of database records 9-54

- subagent, SNMP Multiplex Protocol 7-3

- subsystem definitions, database schema E-5

- subsystem form, modify, Message Definition Editor 9-36

- subsystem name, message attribute, at agent A-4

- SUBSYSTEM, DEFINE\_FILTER 6-11

- SUBSYSTEM, in message definition 5-4

- subsystem\_delete 5-8

- summary, attribute in message definition A-7

- SUMMARY, in message definition 5-4

- support

  - customer xviii

  - documentation xvii

## T

- TALK\_SERVICES, LC\_GLOBAL keyword 6-5

- test messages, generating C-1

- testing and debugging commands C-1

- transaction ID, message attribute, at agent A-5

- trap classes, database schema E-5

- trap enabled, attribute in message definition A-7

- trap generation

  - by message definition and severity, Basic Trap Configuration 9-46

  - by message definition, Basic Trap Configuration 9-45

  - by severity, Basic Trap Configuration 9-43

- trap generation, attribute in message definition A-7

- trap generation, two ways to do it 1-3

- trap ID, attribute in message definition A-8

- TRAP\_ENABLED, in message definition 5-5

- TRAP\_ID, in message definition 5-5

- TRAPID, DEFINE\_FILTER action 6-14

- TUXEDO message manual 5-2

- TXKEY, DEFINE\_FILTER 6-11

## U

- Unacknowledged field, Message Browser 9-5

- user data, database schema E-7

- user ID, message attribute, at agent A-5

- USER, DEFINE\_FILTER 6-11

## W

- Windows NT event log message mapping, predefined G-3

