IBM Tivoli Monitoring
Version 6.2.3 Fix Pack 1

UNIX Logs Agent User's Guide

SC32-9471-05

IBM Tivoli Monitoring
Version 6.2.3 Fix Pack 1

UNIX Logs Agent User's Guide

SC32-9471-05

Note
Before using this information and the product it supports, read the information in "Notices" on page 99.

This edition applies to version 6.2.3 Fix Pack 1 of the IBM Tivoli Monitoring: UNIX Logs Agent (5724-C04) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Tables

Chapter 1. Overview of the Monitoring Agent for UNIX Logs

The Monitoring Agent for UNIX Logs provides you with the capability to monitor UNIX logs and to perform basic actions with UNIX logs. This chapter provides a description of the features, components, and interface options for the Monitoring Agent for UNIX Logs.

IBM Tivoli Monitoring overview

IBM Tivoli Monitoring is the base software for the Monitoring Agent for UNIX Logs. IBM Tivoli Monitoring provides a way to monitor the availability and performance of all the systems in your enterprise from one or several designated workstations.

You can use IBM Tivoli Monitoring to do the following:

- Monitor for alerts on the systems that you are managing by using predefined situations or custom situations.
- Establish your own performance thresholds.
- Trace the causes leading to an alert.
- Gather comprehensive data about system conditions.
- Use policies to perform actions, schedule work, and automate manual tasks.

The Tivoli Enterprise Portal is the interface for IBM Tivoli Monitoring products. By providing a consolidated view of your environment, the Tivoli Enterprise Portal permits you to monitor and resolve performance issues throughout the enterprise.

See the IBM Tivoli Monitoring publications listed in "IBM Tivoli Monitoring library" on page 93 for complete information about IBM Tivoli Monitoring and the Tivoli Enterprise Portal.

Features of the Monitoring Agent for UNIX Logs

On a typical UNIX system, many log files are scattered throughout the file system. The kernel, various utilities, and user applications create these logs to alert an administrator to events such as security violations and software or hardware failures.

As the number of computers that a system administrator oversees increases, the task of managing these logs and utilizing the information they contain becomes increasingly difficult. Additionally, no standardized format exists for UNIX log files; therefore there is no simple way to analyze the data.

Each site handles log management differently and can adopt a strategy that falls between two extremes:

- Discard all log data immediately
- Store all log data indefinitely

While the first choice conserves disk space and the second allows problems to be diagnosed at a later time, neither strategy allows you to anticipate problems or respond to them in a timely manner.

The Monitoring Agent for UNIX Logs allows you to manage and utilize log files more effectively.

- You can create situations that fire when specific messages are written to a log so that you can take a more proactive approach to managing the systems for which you are responsible. This means you can respond to events as soon as they occur and take action to prevent potential problems from developing.
- Because the Monitoring Agent for UNIX Logs screens all log entries forwarding only selected entries to the Tivoli Enterprise Portal, it eliminates the need to manually analyze large log files.
- By shifting the emphasis of management from postmortem diagnosis to real-time response, the monitoring agent allows you to increase the amount of log data collected by system daemons and user applications while decreasing the amount of data archived and stored for historical debugging and analysis.
- You can easily retrieve log entries that occurred within a certain time span from any monitored log. Data from different log types can be presented in a common format within a Tivoli Enterprise Portal workspace.

The Monitoring Agent for UNIX Logs monitors and provides reports for the following types of logs:

- Syslogs
- Utmp style logs
- Errlogs (AIX® platforms only)
- User-defined ASCII logs

User-defined ASCII logs are supported through the Generic User Log Support (GULS) feature. GULS requires that you supply a format command in the configuration file that describes a log's format to the monitoring agent. See Appendix A, "Generic user log support," on page 67 for further details.

Note: The Monitoring for UNIX Logs agent does not support historical data collection or warehousing.

New in this release

For version 6.2.3 of the Monitoring Agent for UNIX Logs, the following enhancements have been made:

- Support for self-describing agents. See the *IBM® Tivoli® Monitoring Installation and Setup Guide* for more information.
- The UNIX_LAA_BP_SysLogError_Critica situation. Note: This predefined situation is based on best practices. While it might not prove perfectly suited to every monitoring environment, it offers a useful starting point for many users.

Monitoring Agent for UNIX Logs components

After you install the Monitoring Agent for UNIX Logs (product code "kul" or "ul") as directed in the *IBM Tivoli Monitoring Installation and Setup Guide*, you have an environment with a client, server, and monitoring agent implementation for IBM Tivoli Monitoring that contains the following components:

- Tivoli Enterprise Portal client with a Java-based user interface for viewing and monitoring your enterprise.
- Tivoli Enterprise Portal Server that is placed between the client and the Tivoli Enterprise Monitoring Server and enables retrieval, manipulation, and analysis of data from the monitoring agents.

- Tivoli Enterprise Monitoring Server, which acts as a collection and control point for alerts received from the monitoring agents, and collects their performance and availability data.
- Monitoring agent, Monitoring Agent for UNIX Logs, which collects and distributes data to a Tivoli Enterprise Monitoring Server.
- Operating system agents and application agents installed on the systems or subsystems you want to monitor. These agents collect and distribute data to the Tivoli Enterprise Monitoring Server.
- Tivoli Data Warehouse for storing historical data collected from agents in your environment. The data warehouse is located on a DB2®, Oracle, or Microsoft SQL database. To collect information to store in this database, you must install the Warehouse Proxy agent. To perform aggregation and pruning functions on the data, install the Warehouse Summarization and Pruning agent.
- Tivoli Enterprise Console event synchronization component for synchronizing the status of situation events that are forwarded to the event server. When the status of an event is updated because of IBM Tivoli Enterprise Console® rules or operator actions, the update is sent to the monitoring server, and the updated status is reflected in both the Situation Event Console and the Tivoli Enterprise Console event viewer. For more information, see *IBM Tivoli Monitoring Installation and Setup Guide*.

User interface options

Installation of the base software and other integrated applications provides the following interfaces that you can use to work with your resources and data:

Tivoli Enterprise Portal browser client interface
The browser interface is automatically installed with Tivoli Enterprise Portal. To start Tivoli Enterprise Portal in your Internet browser, enter the URL for a specific Tivoli Enterprise Portal browser client installed on your Web server.

Tivoli Enterprise Portal desktop client interface
The desktop interface is a Java-based graphical user interface (GUI) on a Windows or a Linux workstation.

IBM Tivoli Enterprise Console
Event management application

Manage Tivoli Enterprise Monitoring Services window
The window for the Manage Tivoli Enterprise Monitoring Services utility is used for configuring the monitoring agent and starting Tivoli services not already designated to start automatically.

Chapter 2. Requirements and configuration for the monitoring agent

This chapter contains information about the following topics and procedures relevant to the installation and configuration of the Monitoring Agent for UNIX Logs:

Requirements for the monitoring agent

In addition to the requirements described in the *IBM Tivoli Monitoring Installation and Setup Guide*, the Monitoring Agent for UNIX Logs has the requirements listed in Table 1.

Table 1. System requirements

Operating system	**UNIX**
Operating system versions	• AIX, v5.2, 5.3 (32-bit or 64-bit) • HP-UX 11i v1 PA-RISC (32-bit) • HP-UX 11i v2 PA-RISC (64-bit) • HP-UX 11i v3 PA-RISC (64-bit) • HP-UX 11i v2 Integrity (64-bit) • HP-UX 11i v3 Integrity (64-bit) • Solaris V8 on SPARC 32-bit (requires Solaris patches 108434-17, 111721-04, and 109147-07) • Solaris V8 on SPARC 64-bit (requires Solaris patches 108435-17, 111721-04, and 108434-17) • Solaris V9 on SPARC 32-bit (requires Solaris patches 111711-11 and 111722-04) • Solaris V9 on SPARC 64-bit (requires Solaris patches 111712-11, 111722-04, and 111711-11) • Solaris V10 on SPARC (32-bit or 64-bit) • Solaris V10 on x86-64 (64-bit) • Linux: – Linux on zSeries - RedHat Enterprise Linux AS 3 (31-bit or 64-bit) - RedHat Enterprise Linux AS 4 (31-bit or 64-bit) - RedHat Enterprise Linux AS 5 (31-bit or 64-bit) - SUSE Linux Enterprise Server 8 (31-bit or 64-bit) - SUSE Linux Enterprise Server 9 (31-bit or 64-bit) - SUSE Linux Enterprise Server 10 (31-bit or 64-bit) – Linux on Intel (32-bit) - RedHat Enterprise Linux AS/ES 3 - RedHat Enterprise Linux AS/ES 4 - RedHat Enterprise Linux AS/ES 5 - SUSE Linux Enterprise Server 8 - SUSE Linux Enterprise Server 9 - SUSE Linux Enterprise Server 10 - RedFlag 4.1 - Asian Linux 2 – Linux on pSeries - RedHat Enterprise Linux AS 4 - RedHat Enterprise Linux AS 5 - SUSE Linux Enterprise Server 9 - SUSE Linux Enterprise Server 10

Table 1. System requirements (continued)

Operating system	**UNIX**
Operating systems versions (continued)	Linux on IA64 (Itanium) • RedHat Enterprise Linux AS 4 [1] • RedHat Enterprise Linux AS 5 [1] • SUSE Linux Enterprise Server 9 [1] • SUSE Linux Enterprise Server 10 [1] • Asian Linux 2.0 Linux on x86-64 • RedHat Enterprise Linux AS 4[1] • RedHat Enterprise Linux AS 5[1] • SUSE Linux Enterprise Server 9[1] • SUSE Linux Enterprise Server 10[1] • Asian Linux 2.0 The Linux version must support the Korn shell (ksh) and Motif Window Manager (libmotif) for installation of the monitoring agent. A POSIX-compliant threads package must be installed on the monitored system.
Memory	• 128 MB RAM at a minimum, 512 MB or higher for better performance
Disk space	• 30 MB of disk space (100 MB for Linux)
Other requirements	• IBM Tivoli Monitoring v6.2.2 agents require at least a v6.2.2 hub monitoring server and portal server. IBM Tivoli Monitoring v6.2.1 hub monitoring servers and portal servers do not support v6.2.2 monitoring agents. IBM Tivoli Monitoring v6.2.1 monitoring agents work with both v6.2.1 and v6.2.2 environments. • TCP/IP • The monitoring agent must have the permissions necessary to perform requested actions. For example, if the user ID you used to log onto the system to install the monitoring agent (locally or remotely) does not have the permission to perform a particular action being monitored by the monitoring agent (such as running a particular command or reading a monitored log file), the monitoring agent will be unable to perform the requested action. • Linux versions require some compatibility libs to be installed for the agent to work correctly. The latest versions of libstdc++, libgcc, compat-libstdc++, and libXp are required for the agent to work correctly. • AIX versions require version 8 of the AIX XL C/C++ runtime. To determine the current level, run the following AIX command: `lslpp -l | grep -i xlc`

Table 1. System requirements (continued)

Operating system	UNIX
Notes: 1. In native 64-bit mode, not tolerance mode.	

Note: For the most current information about the operating systems that are supported, see the following URL: http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html.

When you get to that site, click on the relevant link in the **Operating system reports** section.

Silent installation: If you are performing a silent installation using a response file, see the IBM Tivoli Monitoring Installation and Setup Guide, "Performing a silent installation of IBM Tivoli Monitoring."

Monitoring syslog files on certain AIX 5.3 systems

To monitor the syslog files on a AIX 5.3 system where facility and priority have been added to each message that needs to be logged, complete the following steps:

- Stop the Monitoring Agent for UNIX Logs.
- Insert the **below** entry in the `ul.ini` file available in the directory `$itm_home/config` KUL_NEW_SYSLOG=TRUE.
- Start the Monitoring Agent for UNIX Logs.

To monitor the syslog files on a AIX 5.3 system where a flag, -N, has been added to `syslogd` for not printing the facility and priority information, complete the following steps:

- Stop the Monitoring Agent for UNIX Logs.
- Remove the **below** entry in the `ul.ini` file available in the directory `$itm_home/config` KUL_NEW_SYSLOG=TRUE.
- Start the Monitoring Agent for UNIX Logs.

Specifying the log files to monitor

The runtime environment and behavior of the Monitoring Agent for UNIX Logs is controlled through both a configuration file and environment variables. The configuration file indicates which files the monitoring agent is to monitor.

When the monitoring agent starts, it looks at two files to determine which logs to monitor:

- Customer configuration file
- Syslog daemon configuration file

If, for any reason, the monitoring agent is unable to find at least one log to monitor from either file, it writes a message to the RAS log that contains the text **`Agent has no work to do. Exiting...`** and then automatically terminates.

Customer configuration file

The monitoring agent first looks for a customer configuration file. The absolute or relative file name of this customer configuration file is specified in the monitoring

agent's `ul.ini` file through the KUL_CONFIG_FILE environment variable. (More information about the `ul.ini` file is provided in "Environment variables for the Monitoring Agent for UNIX Logs" on page 10.) The customer configuration file contains a line for each log to be monitored that includes the absolute file name of the log along with the log's type. If the monitoring agent is able to start monitoring at least one of the logs specified in the customer configuration file, it will not interrogate the syslog daemon configuration file.

A default customer configuration file is shipped with the product and is called `kul_configfile`. This file is installed into the *`install_dir`*`/config` directory. All entries in the default file are commented out.

Note: All references to *install_dir* refer to the destination directory that was specified when the monitoring agent was installed.

Customer configuration file format

Each entry in the customer configuration file consists of a single line with the following fields, which must occur in the order given. Each field is delimited by one or more space and/or tab characters and all fields except the first must be preceded by a semicolon (;). There must be no white space between the semicolon and the first character of the field it delimits.

Table 2. Monitoring Agent for UNIX Logs customer configuration file format

Field	Description
1	Absolute file name of monitored log.
2	Debug mode (optional: default = N) N = debug mode off; Y = debug mode on If debug mode is on, each entry written to the monitored log will be recorded in a debug log. In addition, the formatted entry that is passed to a situation is also written to the debug log. All logs that are monitored in debug mode write to the same debug log. The debug log is specified in the monitoring agent `ul.ini` file using the AGENT_DEBUG_LOG environment variable. If this variable is undefined or the log cannot be opened, no debug logging occurs. Each time the monitoring agent is started, new events will be appended to the end of the existing debug log.
3	Log type (optional: default = S) • S = syslog • E = errlog • A = utmp log • U = user-defined log

Table 2. Monitoring Agent for UNIX Logs customer configuration file format (continued)

Field	Description
4	Format command. This command is valid only for type 'E' (errlog) and type 'U' (user-defined) logs. For type 'E' logs, the format command must consist of a an errpt command that includes the '-c' (concurrent mode) option. The default value is: `errpt -c -smmddhhmmyy` For user-defined logs, the format command describes both the format of the log and how data will be mapped and formatted in the Tivoli Enterprise Portal Log Entries table view. There is no default. For additional information on composing format commands, refer "Format command" on page 67.

Syslog daemon configuration file

Kernel daemons and user applications use the UNIX syslog facility to record messages in a log. By using the syslog facility an application ensures that its log entries conform to a standard format.

The actual logging activities are performed by the syslog daemon, `syslogd`, which is controlled through a configuration file usually called `/etc/syslog.conf` or `/etc/syslog_ng.conf` (for the syslog_ng implementation). This file is usually maintained by the system administrator.

The `syslog.conf` or `syslog_ng.conf` file is used to indicate to which syslog messages are to be written that have a given severity and that originate from a given application. This allows you to consolidate messages from more than one source into a single log file. Through the syslog facility, you can also direct messages to be written to a different system, known as the loghost.

The monitoring agent will attempt to build a default list of logs to monitor from the syslog daemon configuration file under the following circumstances:

- The KUL_CONFIG_FILE environment variable is undefined.
- The specified customer configuration file does not exist or cannot be opened.
- There are no log names in the customer configuration file.
- None of the logs contained in the customer configuration file are valid.

The file that the monitoring agent reads to build the default monitored logs list is called `/etc/syslog.conf` or `syslog_ng.conf` (for the syslog_ng implementation), but this can be overridden using the KUL_SYSLOG_CONF environment variable.

If you are interested only in monitoring syslogs, you can omit the KUL_CONFIG_FILE environment variable from the `ul.ini` file, or you can leave the variable unassigned, thereby letting the monitoring agent determine which syslogs are active on each system based on the syslogd configuration file.

Environment variables for the Monitoring Agent for UNIX Logs

Environment variables are specified in the monitoring agent's `ul.ini` file and allow you to communicate to the agent information such as the name of your customer configuration file. The location of the agent's `ul.ini` file is: *`install_dir`*`/config/ul.ini`. The table describes some of the variables you can include.

Table 3. Monitoring Agent for UNIX Logs environment variables (ul.ini file)

Variable name	Purpose
KUL_CONFIG_FILE	The absolute or relative file name of the Monitoring Agent for UNIX Logs customer configuration file. The default value is: *install_dir*/config/kul_configfile
KUL_SYSLOG_CONF	The absolute or relative file name of the syslog daemon configuration file. The default name is: /etc/syslog.conf
AGENT_DEBUG_LOG	The absolute or relative file name of the Monitoring Agent for UNIX Logs debug log. This file is used to record pre- and post-formatted images of each entry written to a log that is being monitored in debug mode. The default value is: *install_dir*/logs/ul_debug.log
KUL_MAX_ROWS	The maximum number of rows to be returned by the Monitoring Agent for UNIX Logs for a Log Entries table view request. The default value is 1000.
KBB_RAS1	Specifies which trace entries to include in the monitoring agent's runtime log. The syntax is: classes (COMP:component classes) (UNIT:unit classes) A class specified outside of a parenthesis is global, that is, it applies to all components and units. Valid classes are: • ERROR • FLOW • STATE • DETAIL • ALL Including one or more classes within parentheses includes entries of the class that are generated by the associated component or unit. A useful component to trace is "kul"; tracing units of "kul" and/or "kra" can also be informative.
CTIRA_HOSTNAME	Overrides the name with which the monitoring agent identifies itself to the server. The default is the computer name.
CTIRA_NODETYPE	Overrides the default suffix that is appended to the host name to differentiate the Monitoring Agent for UNIX Logs from another UNIX monitoring agent running on the same computer. The default is KUL.

Environment variable syntax

The syntax for defining an environmental variable depends on the shell used to interpret the script. In the Bourne and Korn shells, the following defines the variable "VAR" assigning to it the value "VALUE" and makes it available to other programs invoked subsequently in the same script:

```
VAR=VALUE; export VAR
```

If you are using the C shell, the following command would produce the same result:

```
setenv VAR VALUE
```

Examples

If you are using the Bourne or Korn shells, use the following commands to assign a value of *install_dir*/config/myconfig to the KUL_CONFIG_FILE variable.

```
KUL_CONFIG_FILE=install_dir/config/myconfig
export KUL_CONFIG_FILE
```

If you're using the C shell, use the following command:

```
setenv KUL_CONFIG_FILE install_dir/config/myconfig
```

Dynamically refreshing the monitoring agent

After the monitoring agent starts, you can dynamically change the list of logs being monitored on managed systems; it is not necessary to stop and restart the monitoring agent.

In addition, if one or more monitors were either unable to start, or terminated abnormally, they can be restarted without the need to restart the monitoring agent. In both cases, it is only necessary to send the monitoring agent a refresh signal.

Sending a refresh signal to the monitoring agent

Use the following procedure to dynamically change the list of logs being monitored on a managed system, or to restart individual monitors.

1. Start a Telnet session or other remote login procedure to the managed system on which you want to change the monitored logs list.
2. Modify the customer configuration file (if you have specified the KUL_CONFIG_FILE environment variable) or the syslog daemon configuration file.
3. Send the monitoring agent a refresh signal:

   ```
   kill -HUP agentPID
   ```
4. Open the Monitored Logs table view for the appropriate managed system. The result is that monitoring has begun for logs that were added to the configuration file, and stopped for logs that were deleted from the configuration file.

Note: A refresh occurs only if the monitoring agent determines that the configuration file has been modified since the agent was started, or since the previous refresh. If you have not modified the configuration file, but want to restart a monitor, change the modification date of the configuration file prior to sending a refresh signal.

To change the modification date of the configuration file prior to sending a signal, issue the following command from the *install_dir*/config directory on the managed system where the monitoring agent is running:

```
touch kul_configfile
```

Generic User Log Support (GULS)

The Monitoring Agent for UNIX Logs has built into it the ability to monitor three types of standard UNIX logs: syslogs, errlogs, and utmp logs. To monitor a log of one of these three types, it is only necessary to specify the name and type of the log in the configuration file because the monitoring agent already understands how to interpret the data within each log record and map it into the Log Entries table view.

To monitor an ASCII log that does not conform to any of the three supported types, see Appendix A, "Generic user log support," on page 67.

Running as a non-Administrator user

The Monitoring Agent for UNIX Logs runs as a setuid root program. The primary reason for this is that the agent monitors log files that are potentially owned by a variety of users, and those logs might not have the correct permissions to allow other users access. Running as root, the Monitoring Agent for UNIX Logs is able to monitor any log the customer selects. This root authority can be removed from the Monitoring Agent for UNIX Logs if the appropriate permissions are set on the log files to allow the agent to monitor them.

Setting up the Monitoring Agent for UNIX Logs in a cluster environment

The *IBM Tivoli Monitoring Installation and Setup Guide* contains an overview of clustering. The information provided here is specifically for installing and setting up the Monitoring Agent for UNIX Logs in a Microsoft Cluster Server environment.

The Monitoring Agent for UNIX Logs is set up and works as it would in a non-clustered environment for those log files stored on the local disk.

Chapter 3. Workspaces reference

This chapter contains an overview of workspaces, references for detailed information about workspaces, and descriptions of the predefined workspaces included in this monitoring agent.

About workspaces

A workspace is the working area of the Tivoli Enterprise Portal application window. At the left of the workspace is a Navigator that you use to select the workspace you want to see.

As you select items in the Navigator, the workspace presents views pertinent to your selection. Each workspace has at least one view. Some views have links to workspaces. Every workspace has a set of properties associated with it.

This monitoring agent provides predefined workspaces. You cannot modify or delete the predefined workspaces, but you can create new workspaces by editing them and saving the changes with a different name.

More information about workspaces

For more information about creating, customizing, and working with workspaces, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the predefined workspaces for this monitoring agent and a description of each workspace, refer to the Predefined workspaces section below and the information in that section for each individual workspace.

Predefined workspaces

The Monitoring Agent for UNIX Logs provides the following predefined workspaces:

- Log Entries
- Monitored Logs

The Log Entries workspace uses the link from the Monitored Logs workspace to pass the required parameters. If you wish to create a custom workspace, for example, one that shows the entries from two logs, you must create a custom query for each table that has the following formula:

```
(Managed System == $NODE$ AND Log Path (Unicode) == path AND Log Name (Unicode)
== name  AND Entry Time >= start_time AND Entry Time <= end_time)
```

The values in italics are the desired parameters from the user.

Note: To create this query, you must add two Entry Time attributes in order to create the AND operation described above. If you simply enter two values for a single Entry Time attribute in the Tivoli Enterprise Portal Query Editor, you will, instead create an OR relationship.

Some predefined workspaces are not available from the Navigator tree item, but are accessed by selecting the link indicator next to a row of data in a view.

Left-clicking a link indicator selects the default workspace associated with that link. Right-clicking a link indicator displays all linked workspaces that can be selected.

The remaining sections of this chapter contain descriptions of each of these predefined workspaces.

Log Entries workspace

The Log Entries workspace displays entries from any monitored log that occurred within a specified time range. The same format is used for all logs, regardless of their type. After the Tivoli Enterprise Portal has retrieved and displayed the entries for the required time range, you can perform additional sorting and filtering. This workspace is comprised of three views:

- Log Entries (table view)
- Log Size (bar chart)
- Number of Events (bar chart)

The Log Entries table view provides entry data and a description of each entry in the monitored log. The Log Size chart depicts the size of each monitored log file, in bytes. The Number of Events chart depicts the total number of events detected by the monitor since the monitor was first started. Based on the information that this workspace provides, you can make changes, and set up situations.

Monitored Logs workspace

The Monitored Logs workspace provides basic information about the logs you are monitoring. Workspace columns display:

- Logs that you have elected to monitor
- Basic information about each log, such as the log size and the time at which the last log was modified
- Status of each monitor
- Time at which each monitor started or stopped
- Number of events detected by each monitor

This workspace is comprised of three views:

- Log Size (bar chart)
- Monitored Logs (table view)
- Number of Events (bar chart)

The Log Size chart depicts the size of each monitored log file, in bytes. The Monitored Logs table view lists a variety of status details associated with the logs you are monitoring. The Number of Events chart depicts the total number of events detected by the monitor since the monitor was first started. Based on the information that this workspace provides, you can make changes, and set up situations.

Typical scenarios

This section illustrates how you can use the workspaces to monitor logs in some typical scenarios.

Security issues

A common technique used by hackers to gain unauthorized access to your systems is to guess the password for a known userid, often for the superuser. Failed logon

attempts are often recorded in a log. For instance, if someone issues the "su" command to change the user ID to which they are currently logged on and enters an invalid password for the new user ID, an entry is usually written to the file `/usr/adm/suaudit`. You can create a situation that alerts you to possible break-in attempts when repeated login failures to a user ID occur within a short period of time.

Display the Monitored Logs workspace for the system in question to confirm that you are monitoring the appropriate log. Display the Log Entries workspace for the log to verify the format of the message written when a login attempt fails, for example, "BAD SU from user1 to root". Construct a situation that will fire if a message indicating a logon failure to "root" is detected more frequently than would normally be expected.

File server problems

To enable the sharing of data between multiple systems, many sites designate one system as a file server and utilize a facility such as NFS to service remote file access requests. In these environments, network outages or problems with the server itself can impact many users. Often, a message is written to a log, for example, `/var/adm/messages`, if an NFS request issued by a client on a remote system fails.

Display the Monitored Logs workspace for a client system to confirm that you are monitoring the appropriate log. Display the Log Entries workspace for the log to verify the format of the message written when an NFS request fails, for example, "NFS server system1 not responding". Construct a situation that fires if a message indicating a server problem is detected more frequently then would normally be expected.

Monitoring user and third-party vendor applications

You can monitor any application that is already writing messages to one of the supported standard log types (for example, syslog, utmp and errlog), by simply adding an entry for the log in configuration file and restarting or refreshing the monitoring agent.

If you want to monitor an application that is not already writing to a log file and you can modify the application, add the necessary syslog system calls (or, on AIX platforms, errlog system calls), to the code to produce the desired messages and add the new logs to the configuration file.

If you want to monitor an application that you cannot modify and which is writing messages to an ASCII log in a non-standard format, add the log to the configuration file as normal but set its type to 'U' (user-defined). User-defined logs require a format command that the monitoring agent uses to read log entries and map the data into the Log Entries workspace.

Suppose you wish to monitor a log called `/usr/adm/logs/mylog` that is comprised of entries such as those following:

```
MSG123456I/1024 10/04/05 12:15:32 region1 : Application 8 started
MSG234567W/2048 10/04/05 13:01:31 region2 : No journal files
opened
MSG345678E/4096 10/04/05 14:57:02 region1 : Unable to open file
'FILE1'
```

The message identifier is terminated by either 'I' (informational), 'W' (warning) or 'E' (Error) depending on the severity of the message. You want to create a situation that will fire only when type 'E' messages are written. In the Log Entries

workspace for this log, you want the message identifier to prefix the message text and both to be displayed in the description column. Lastly, the decimal number that is displayed after the message identifier is a reason code that you wish to convert to hexadecimal and display in the class column prefixed by the literal “RC =“.

The following is the entry that you would include in the configuration file that will format the log entries appropriately both for your situation and for report requests. See “Customer configuration file format” on page 9 for details on the format of the configuration file.

Note: The entire entry must be contained on a single line.

```
/usr/adm/logs/mylog ;N ;U ;a,"%9s%c/%d %d/%d/%d %d:%d:%d %s
:%[^\n]" , desc type class = "RC = %x" month day year hour
min sec source desc = " %s"
```

Analyzing this line one component at a time from left to right, you can see that the first item specifies the absolute file name of the log you want to monitor, in this case `/usr/adm/logs/mylog`. Following the white space delimiter and the semicolon (which indicates the start of the next item), is an ‘N’ stating that this log is not being monitored in debug mode.

The ‘U’ item indicates that the log type is user-defined which mandates the presence of the last item, the format command.

The format command starts with an ‘a’ (ASCII) and a comma. Everything between the double quotes that follow is part of the format description that tells the monitoring agent the format of the log. This format description allows you to break each log entry into arbitrary fields consisting of one or more characters. Following the format description is a comma that is followed by the mapping specifications. These indicate into which column of the Log Entries workspace each field must be mapped.

In this example, the format description breaks each log entry into 11 fields. Each field is then mapped into one column of the Log Entries workspace by the 11 mapping specifiers.

Table 4. Log entry fields and their descriptions

Field	Scan directive	Mapping and formatting data into the Log Entries workspace
1	%9s	Consumes the first 9 characters of the message identifier and is mapped into the “description” column.
2	%c	Consumes the 10th character, in this example the message severity indicator, (I, W or E). This is mapped into the “type” column.
3	/%d	Consumes the ‘/’ character and the following integer. The ‘/’ literal causes the ‘/’ character to be discarded but the integer is mapped into the ‘class’ column. The mapping specification for the class column includes a format specifier “RC=%x”. This inserts the literal “RC =” into the column and converts the integer to hexadecimal.
4	%d	Consumes the white space and the following integer mapping it into the “month” component of the Entry Time column.

Table 4. Log entry fields and their descriptions (continued)

Field	Scan directive	Mapping and formatting data into the Log Entries workspace
5	/%d	Consumes and discards the '/' character and then consumes the following integer mapping it into the "day" component of the Entry Time column.
6	/%d	Consumes and discards the '/' character; then consumes the following integer mapping it into the "year" component of the Entry Time column.
7	%d	Consumes the next integer mapping it into the 'hour' component of the Entry Time column.
8	:%d	Consumes and discards the ':' character and then consumes the following integer mapping it into the 'minute' component of the Entry Time column.
9	:%d	Consumes and discards the ':' character and then consumes the following integer mapping it into the 'second' component of the Entry Time column.
10	%s	Consumes and discards the white space preceding the next character; then maps the next character string, (terminated by white space), into the 'source' column.
11	:%[^\n]	Consumes and discards all white space preceding the colon and the colon itself in a log entry. The scanset discards all white space between the colon and message content in a log entry and then maps all remaining characters in the entry into the 'description' column. Since the message identifier is also being mapped into this same column, this mapping specification includes a format specifier, " %s", which simply inserts a space between the two mapped fields.

The resulting Log Entries workspace is displayed as follows:

Table 5. Sample Log Entry workspace

Entry time	Description	Source	System	Class	Type
10/04/05 14:57:02	MSG345678 Unable to open file 'FILE1'	region1		RC = 1000	E
10/04/05 13:01:3	MSG234567 No journal files opened	region2		RC = 800	W
10/04/05 12:15:32	MSG123456 Application 8 started	region1		RC = 400	I

You can now create a situation that fires if any 'E' type messages are written to this log file by including the following predicates:

- Log_Entries.Log_Name= mylog
- Log_Entries.Type= E

Resetting a situation using the Until predicate

When fired, situations that are based on the Log Entries workspace will remain in a raised state. From the Events View, you have the option to reset a situation that has fired thereby changing the managed object's state back to the normal OK state.

You can also include an 'Until' predicate in the situation that causes it to be reset automatically. The Until predicate allows you to specify that the situation is to be reset after a certain time interval or when another situation is true.

This can be useful if you are monitoring events that occur in pairs, for example:

```
server redwood not responding
server redwood OK
```

In this case, you might create a situation called "Server_OK" that monitors a certain log file looking for messages that contain the text "redwood OK". Then create another situation called "Server_Error" monitoring the same log but looking for the text "redwood not responding". In this second situation, include an Until predicate. Open the Until settings page and refer to the "Reset this situation when" box that contains 3 radio buttons. Select the button called "Another situation is TRUE" and in the "Resetting situation" box enter the name of the first situation, "Server_OK".

Distribute both situations as usual to the managed systems on which you want them to run. Add the situation called "Server Error" to one of the states of a new or existing template and drag the template to the Enterprise icon to create a managed object. Assign the managed object to one or more of the managed systems to which you distributed the situations

When a event is written to the monitored log containing the text "redwood not responding", the Server_Error situation fires causing the managed object to change state. When a message is written to the log containing the text "redwood OK", the Server_Error situation is reset and the managed object state reverts to normal.

Chapter 4. Attributes reference

This chapter contains information about the following topics:

- Overview of attributes
- References for detailed information about attributes
- Descriptions of the attributes for each attribute group included in this monitoring agent

About attributes

Attributes are the application properties being measured and reported by the Monitoring Agent for UNIX Logs, such as the log size. Some monitoring agents have fewer than 100 attributes, while others have over 1000.

Attributes are organized into groups according to their purpose. The attributes in a group can be used in the following two ways:

- Chart or table views

  Attributes are displayed in chart and table views. The chart and table views use queries to specify which attribute values to request from a monitoring agent. You use the Query editor to create a new query, modify an existing query, or apply filters and set styles to define the content and appearance of a view based on an existing query.
- Situations

  You use attributes to create situations that monitor the state of your operating system, database, or application. A situation describes a condition you want to test. When you start a situation, the Tivoli Enterprise Portal compares the values you have assigned to the situation attributes with the values collected by the Monitoring Agent for UNIX Logs and registers an *event* if the condition is met. You are alerted to events by indicator icons that appear in the Navigator.

Some of the attributes in this chapter are listed twice, with the second attribute having a "(Unicode)" designation after the attribute name. These Unicode attributes were created to provide access to globalized data.

More information about attributes

For more information about using attributes and attribute groups, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the attributes groups, a list of the attributes in each attribute group, and descriptions of the attributes for this monitoring agent, refer to the Attribute groups and attributes section in this chapter.

Attribute groups and attributes for the Monitoring Agent for UNIX Logs

This monitoring agent contains the following attribute groups:

- Log Entries Attributes
- Monitored Logs Attributes

The following sections contain descriptions of these attribute groups, which are listed alphabetically. Each description contains a list of attributes in the attribute group.

IBM Tivoli Monitoring provides other attribute groups that are available to all monitoring agents, for example Universal Time and Local Time. The attributes in these common attribute groups are documented in the Tivoli Enterprise Portal Help.

Log Entries Attributes

Use the Log Entries attributes to create situations, except the Managed System attribute, to monitor entries made to monitored logs.

Class The class of entry for errlogs, indicated by S = Software, H = Hardware, or O = Error Logger. Valid entry is an alphanumeric text string, with a maximum length of 16 characters.

Description The content of the log entry. Valid entry is an alphanumeric text string, with a maximum length of 256 characters.

Description (Unicode) The content of the log entry. Valid entry is a text string, with a maximum length of 768 bytes.

Entry Time The date and time, as set on the monitored system, indicating the instance when the entry was written. The timestamp format for SCAN and STR functions is CYYMMDDHHMMSSmmm (as in 1020315064501000 for 03/15/02 06:45:01) where:

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

Milliseconds are not used and are always be 000.

Frequency Threshold The number of times an event must occur within a user-specified interval before a situation is raised. Valid entry is an integer of up to four bytes. Note that Frequency Threshold does not display in the workspace, although it can be used as a situation predicate.

Log Name The name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 128 characters.

Log Name (Unicode) The name of the monitored log. Valid entry is a text string, with a maximum length of 384 bytes.

Log Path The absolute path name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 256 characters.

Log Path (Unicode) The absolute path name of the monitored log. Valid entry is a text string, with a maximum length of 768 bytes.

Managed System The managed system name. The form should be *hostname:agent_code*.

Examples include spark:KUL or deux.raleigh.ibm.com:KUL.

In workspace queries, this attribute should be set equal to the value $NODE$ in order to populate the workspace with data. This attribute is generally not included in situations, unless there is a need to customize the situation for a specific managed system.

Period Threshold The interval in seconds within which an event must occur at least a user-specified number of times before a situation is raised. Valid entry is an integer of up to four bytes. Note that Period Threshold does not display in the workspace, although it can be used as a situation predicate.

Source The application or resource that logged the entry. Valid entry is an alphanumeric text string, with a maximum length of 32 characters.

Source (Unicode) The application or resource that logged the entry. Valid entry is a text string, with a maximum length of 96 bytes.

System The system on which the entry was written. Valid entry is an alphanumeric text string, with a maximum length of 32 characters.

Timestamp The date and time, as set on the monitored system, indicating the instance when the agent collects information. The timestamp format for SCAN and STR functions is CYYMMDDHHMMSSmmm (as in 1020315064501000 for 03/15/02 06:45:01) where:

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

Milliseconds are not used and are always be 000.

Type The type of entry for errlogs and utmp logs. For errlog entries, the entry types include "P" (for "PERM, "PERF" and "PEND"), "T"(for TEMP), "I" (For INFO) and "U" (For UNKN) as the Monitoring Agent for UNIX Logs supports only the summary report of the errpt command. For utmp logs, the entry types include Unused space, Run level, System boot time, User logon time, User idle time, init process, getty waiting, User process, Zombie process, and Accounting. Valid entry is an alphanumeric text string, with a maximum length of 16 characters.

Monitored Logs Attributes

Use the Monitored Logs attributes to create situations to monitor logs on a remote node.

Date Last Modified The date and time, as set on the monitored system, indicating the instance when the log was last modified. The timestamp format for SCAN and STR functions is CYYMMDDHHMMSSmmm (as in 1020315064501000 for 03/15/02 06:45:01) where:

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

Debug Mode This attribute indicates whether or not preformatted and postformatted events written to this log are also written to a debug log. Valid entry is an alphanumeric text string, with a maximum length of one character. N for no and Y for yes are the two possible values.

Format Command The name of the command to be invoked to format log entries. For errlogs, this attribute represents the name of the command to be invoked to format entries in ASCII format. For user logs, this attribute represents the format command used to describe the log's format and how the data should be mapped and formatted in the Log Entries workspace. Valid entry is an alphanumeric text string, with a maximum length of 256 characters.

Log Name The name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 128 characters.

Log Name (Unicode) The name of the monitored log. Valid entry is a text string, with a maximum length of 384 bytes.

Log Path The absolute path name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 256 characters.

Log Path (Unicode) The absolute path name of the monitored log. Valid entry is a text string, with a maximum length of 768 bytes.

Log Size (Bytes) The size of the monitored log file, in bytes. Valid entry is an integer in the range zero to 9223372036854775807. Note: the value -2 indicates Not Collected and the value 9223372036854775807 indicates Value_Exceeds_Maximum.

Log Size (Bytes) (Superseded) The size of the monitored log file, in bytes. Valid entry is an integer of up to four bytes, and the range is between 0 and 2147483647. Note: the value -2 indicates Not Collected and the value 2147483647 indicates Value_Exceeds_Maximum.

Log Type The log type, indicated by A = Administrative Log, E = Error Log, S = System Log, or U = User Log. Valid entry is an alphanumeric text string, with a maximum length of one character.

Managed System The managed system name. The form should be *hostname*:*agent_code*.

Examples include spark:KUL or deux.raleigh.ibm.com:KUL.

In workspace queries, this attribute should be set equal to the value $NODE$ in order to populate the workspace with data. This attribute is generally not included in situations, unless there is a need to customize the situation for a specific managed system.

Monitor Start/Stop Time A timestamp indicating the time at which a monitor started running (if the monitor status is running) or the time at which the monitor terminated. The timestamp format for SCAN and STR functions is CYYMMDDHHMMSSmmm (as in 1020315064501000 for 03/15/02 06:45:01) where:

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

Milliseconds are not used and are always be 000.

Monitor Status If the log monitor is active, the status will be running; otherwise, the status will indicate the error that caused the monitor to terminate. Valid entry is an alphanumeric text string, with a maximum length of 32 characters. The following values are valid:

- Error: create child failed
- Error: create pipe failed

- Error: format command
- Error: get pipe flag failed
- Error: insufficient memory
- Error: log open failed
- Error: log read failed
- Error: log rewind failed
- Error: pipe FD to FP failed
- Error: reset EOF failed
- Error: seek for EOF failed
- Error: set pipe flag failed
- Error: unknown
- Error: wait for event failed
- Error: wait loop failed
- Not started
- Running
- Stopped

Number of Events The number of events detected by the monitor since monitoring started. Valid entry is an integer in the range zero to 9223372036854775807. Valid values can include the value Value_Exceeds_Maximum=9223372036854775807.

Number of Events (Superseded) The number of events detected by the monitor since monitoring started. Valid entry is an integer of up to four digits, and the range is between 0 and 2147483647. Valid values can include the value Value_Exceeds_Maximum=2147483647.

Number of Format Errors The number of events that the monitor was unable to understand and format (and as a result, were discarded). Valid entry is an integer in the range zero to 9223372036854775807. Valid values can include the value Value_Exceeds_Maximum=9223372036854775807.

Number of Format Errors (Superseded) The number of events that the monitor was unable to understand and format (and as a result, were discarded). Valid entry is an integer of up to four bytes, and the range is between 0 and 2147483647. Valid values can include the value Value_Exceeds_Maximum=2147483647.

Timestamp The date and time, as set on the monitored system, indicating the instance when the agent collects information. The timestamp format for SCAN and STR functions is CYYMMDDHHMMSSmmm (as in 1020315064501000 for 03/15/02 06:45:01) where:

C = Century (0 for 20th, 1 for 21st)

Y = Year

M = Month

D = Day

H = Hour

M = Minute

S = Second

m = millisecond

Milliseconds are not used and are always be 000.

Chapter 5. Situations reference

This chapter contains an overview of situations, references for detailed information about situations, and descriptions of the predefined situations included in this monitoring agent.

About situations

A situation is a logical expression involving one or more system conditions. Situations are used to monitor the condition of systems in your network. You can manage situations from the Tivoli Enterprise Portal by using the Situation editor.

The IBM Tivoli Monitoring agents that you use to monitor your system environment are shipped with a set of predefined situations that you can use as-is or you can create new situations to meet your requirements. Predefined situations contain attributes that check for system conditions common to many enterprises.

Using predefined situations can improve the speed with which you can begin using the Monitoring Agent for UNIX Logs. You can examine and, if necessary, change the conditions or values being monitored by a predefined situation to those best suited to your enterprise.

Note: The predefined situations provided with this monitoring agent are not read-only. Do not edit these situations and save over them. Software updates will write over any of the changes that you make to these situations. Instead, clone the situations that you want to change to suit your enterprise.

You can display predefined situations and create your own situations using the Situation editor. The left frame of the Situation editor initially lists the situations associated with the Navigator item that you selected. When you click a situation name or create a new situation, the right frame opens with the following tabs:

Formula
Condition being tested

Distribution
List of managed systems (operating systems, subsystems, or applications) to which the situation can be distributed.

Expert Advice
Comments and instructions to be read in the event workspace

Action
Command to be sent to the system

Until Duration of the situation

More information about situations

The *IBM Tivoli Monitoring User's Guide* contains more information about predefined and custom situations and how to use them to respond to alerts.

For a list of the predefined situations for this monitoring agent and a description of each situation, refer to the Predefined situations section below and the information in that section for each individual situation.

Predefined situations

This monitoring agent contains the following predefined situations, which are organized alphabetically:

- HACMP_acquire_service_addr
- HACMP_acquire_takeover_ addr
- HACMP_config_too_long
- HACMP_event_error
- HACMP_fail_standby
- HACMP_get_disk_vg_fs
- HACMP_join_standby
- HACMP_network_down
- HACMP_network_down_ complete
- HACMP_network_up
- HACMP_network_up_complete
- HACMP_node_down
- HACMP_node_down_complete
- HACMP_node_down_local
- HACMP_node_down_local_ complete
- HACMP_node_down_remote
- HACMP_node_down_rmt_ complete
- HACMP_node_up
- HACMP_node_up_complete
- HACMP_node_up_local
- HACMP_node_up_local_ complete
- HACMP_node_up_remote
- HACMP_node_up_remote_ complete
- HACMP_release_service_addr
- HACMP_release_takeover_addr
- HACMP_release_vg_fs
- HACMP_start_server
- HACMP_stop_server
- HACMP_swap_adapter
- HACMP_swap_adapter_complete
- UNIX_LAA_Bad_su_to_root_Warning
- UNIX_LAA_BP_SysLogError_Critica
- UNIX_LAA_Log_Size_Warning
- UNIX_LAA_Log_Size_Warning_2

The situations for High Availability Cluster Multiprocessing (HACMP™) are supported only on the AIX platform.

The remaining sections of this chapter contain descriptions of each of these predefined situations. The situations are organized alphabetically.

HACMP_acquire_service_addr situation

Configures the boot address to the corresponding service address and starts TCP/IP servers and network daemons by running the **telinit a** command.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ acquire_service_addr
```

HACMP_acquire_takeover_ addr situation

Acquires the takeover IP address by checking configured standby addresses and swapping them with failed service addresses.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ acquire_takeover_addr
```

HACMP_config_too_long situation

Sends a periodic console message when a node has been in reconfiguration for more than six minutes.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ config_too_long
```

HACMP_event_error situation

Occurs when an HACMP event script fails for some reason.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ event_error
```

HACMP_fail_standby situation

Sends a console message when a standby adapter fails or is no longer available because it has been used to take over the IP address of another adapter.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ fail_standby
```

HACMP_get_disk_vg_fs situation

Acquires disk, volume group, and file system resources as part of takeover.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ get_disk_vg_fs
```

HACMP_join_standby situation

Sends a console message when a standby adapter becomes available.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ join_standby
```

HACMP_network_down situation

Occurs when the cluster determines that a network has failed. The event script provided takes no default action because the appropriate action is site or LAN specific.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ network_down
```

HACMP_network_down_complete situation

Occurs only after a network_down event has successfully completed. The event script provided takes no default action because the appropriate action is site or LAN specific.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ network_down_complete
```

HACMP_network_up situation

Occurs when the cluster determines that a network has become available. The event script provided takes no default action because the appropriate action is site or LAN specific.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ network_up
```

HACMP_network_up_complete situation

Occurs only after a network_up_event has successfully completed. The event script provided takes no default action because the appropriate action is site or LAN specific.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ network_up_complete
```

HACMP_node_down situation

Occurs when a node is detaching from the cluster, either voluntarily or because of a failure. Depending on whether the node is local or remote, either the node_down_local or node_down_remote subevent is called.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ node_down
```

HACMP_node_down_complete situation

Occurs only after a node_down event has successfully completed. Depending on whether the node is local or remote, either the node_down_local or node_down_remote_complete subevent is called.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ node_down_complete
```

HACMP_node_down_local situation

Releases resources taken from a remote node, stops application servers, releases a service address taken from a remote node, releases concurrent volume groups, unmounts file systems, and reconfigures the node to its boot address.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_down_local
```

HACMP_node_down_local_complete situation

Instructs the cluster manager to exit when the local node has completed detaching from the cluster. This event occurs only after a node_down_local event has successfully completed.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ  node_down_local_complete
```

HACMP_node_down_remote situation

Unmounts any NFS file systems and places a concurrent volume group in nonconcurrent mode when the local node is the only surviving node in the cluster. If the failed node did not go down gracefully, acquires failed nodes resources: file systems, volume groups, and disks and service address.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log  *AND *SCAN Log_Entries.Description
*EQ node_down_remote
```

HACMP_node_down_rmt_complete situation

Starts takeover application servers if the remote node did not go down gracefully. This event occurs only after node_down_remote event has successfully completed.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_down_remote_complete
```

HACMP_node_up situation

Occurs when a node is joining the cluster. Depending on whether the node is local or remote, either the node_up_local or node_up_remote subevent is called.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ node_up
```

HACMP_node_up_complete situation

Occurs only after a node_up has successfully completed. Depending on whether the node is local or remote, either node_up_local_complete or node_up_remote_complete subevent is called.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log  *AND *SCAN Log_Entries.Description
*EQ node_up_complete
```

HACMP_node_up_local situation

When the local node attaches to the cluster, the HACMP_node_up_local situation acquires the services address, clears the application server file, acquires file systems, volume groups and disk resources, exports file systems and either activates concurrent volume groups or puts them into concurrent mode depending on the status of the remote nodes.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_up_local
```

HACMP_node_up_local_complete situation

Starts application servers and then checks to see if an inactive takeover is needed. This event only occurs after node_up_local event has successfully completed.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_up_local_complete
```

HACMP_node_up_remote situation

Causes the local node to do an NFS mount only after the remote node starts and to place the concurrent volume groups into concurrent mode.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_up_remote
```

HACMP_node_up_remote_complete situation

Allows the local node to do an NFS mount only after the remote node is completely up. This event occurs only after a node_up_remote event has successfully completed.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ node_up_remote_complete
```

HACMP_release_service_addr situation

Detaches the service address and reconfigures to the boot address.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ release_service_addr
```

HACMP_release_takeover_addr situation

Identifies a takeover address to be released because a standby adapter on the local node is masquerading as the service address of the remote node. Reconfigures the local standby into its original role.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ release_takeover_addr
```

HACMP_release_vg_fs situation

Releases volume groups and file systems that the local node took from the remote node.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ release_vg_fs
```

HACMP_start_server situation

Starts application servers.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ start_server
```

HACMP_stop_server situation

Stops application servers.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log*AND *SCAN Log_Entries.Description
*EQ stop_server
```

HACMP_swap_adapter situation

Exchanges or swaps the IP addresses of two network interface. NIS and name serving are temporarily turned off during the event.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ swap_adapter
```

HACMP_swap_adapter_complete situation

Occurs only after a swap_adapter event has successfully completed. Ensures that the local Address Resolution Protocol (ARP) cache is updated by deleting entries and pinging cluster IP addresses.

This situation is not activated at startup.

This situation has the following formula:

```
*IF *SCAN Log_Entries.Log_Name *EQ cluster.log *AND *SCAN Log_Entries.Description
*EQ swap_adapter_complete
```

UNIX_LAA_Bad_su_to_root_Warning situation

Raises an alert if a logon failure to root message is written to usr/adm/suaudit more than three times within a minute.

This situation has the following formula:

```
Log_Entries.Log_Path EQ /usr/adm/
AND
Log_Entries.Log_Name EQ suaudit
AND
*SCAN Log_Entries.Description EQ 'Badsu'
AND
```

```
*SCAN Log_Entries.Description EQ 'to root' AND
Log_Entries.Frequency_Threshold GT 3
AND
Log_Entries.Period_Threshold EQ 60
```

UNIX_LAA_BP_SysLogError_Critica situation

Monitors log files for entries containing the string 'error.' This situation is not automatically distributed during installation to the default MSL or managed system.

This situation has the following formula:

```
*IF ( ( *VALUE Log_Entries.Log_Path_U *EQ '/var/adm' *AND *VALUE Log
_Entries.Log_Name_U *EQ 'messages' *AND *SCAN Log_Entries.Description_U
*EQ 'error' *AND *SCAN Log_Entries.Description_U *NE 'PAM' ) *OR ( *VALU
E Log_Entries.Log_Path_U *EQ '/var/log' *AND *VALUE Log_Entries.Log_Name
_U *EQ 'messages' *AND *SCAN Log_Entries.Description_U *EQ 'error' *AND
*SCAN Log_Entries.Description_U *NE 'pam_ldap' *AND *SCAN Log_Entries.De
scription_U *NE 'PAM' ) )
```

UNIX_LAA_Log_Size_Warning situation

This situation has been superseded by UNIX_LAA_Log_Size_Warning_2. Raises an alert if the size of any monitored log exceeds 10 MB.

This situation has the following formula:

```
Monitored_Logs.Size GT 10485760
```

UNIX_LAA_Log_Size_Warning_2 situation

Raises an alert if the size of any monitored log exceeds 10 MB.

This situation has the following formula:

```
Monitored_Logs.Size GT 10485760
```

Chapter 6. Take Action commands reference

This chapter contains an overview of Take Action commands and references for detailed information about Take Action commands.

About Take Action commands

Take Action commands can be run from the desktop or included in a situation or a policy.

When included in a situation, the command executes when the situation becomes true. A Take Action command in a situation is also referred to as reflex automation. When you enable a Take Action command in a situation, you automate a response to system conditions. For example, you can use a Take Action command to send a command to restart a process on the managed system or to send a text message to a cell phone.

Advanced automation uses policies to perform actions, schedule work, and automate manual tasks. A policy comprises a series of automated steps called activities that are connected to create a workflow. After an activity is completed, Tivoli Enterprise Portal receives return code feedback, and advanced automation logic responds with subsequent activities prescribed by the feedback.

More information about Take Action commands

For more information about working with Take Action commands, see the *IBM Tivoli Monitoring User's Guide*.

Predefined Take Action commands

There are no predefined Take Action commands for this monitoring agent; however, you can run commands yourself, and include those that you use often in a list of available commands.

Chapter 7. Policies reference

This chapter contains an overview of policies and references for detailed information about policies.

About policies

Policies are an advanced automation technique for implementing more complex workflow strategies than you can create through simple automation.

A *policy* is a set of automated system processes that can perform actions, schedule work for users, or automate manual tasks. You use the Workflow Editor to design policies. You control the order in which the policy executes a series of automated steps, which are also called activities. Policies are connected to create a workflow. After an activity is completed, Tivoli Enterprise Portal receives return code feedback and advanced automation logic responds with subsequent activities prescribed by the feedback.

Note: For monitoring agents that provide predefined policies, predefined policies are not read-only. Do not edit these policies and save over them. Software updates will write over any of the changes that you make to these policies. Instead, clone the policies that you want to change to suit your enterprise.

More information about policies

For more information about working with policies, see the *IBM Tivoli Monitoring User's Guide*.

For information about using the Workflow Editor, see the *IBM Tivoli Monitoring Administrator's Guide* or the Tivoli Enterprise Portal online help.

Predefined policies

There are no predefined policies for this monitoring agent.

Chapter 8. Troubleshooting

This appendix explains how to troubleshoot the Monitoring Agent for UNIX Logs. Troubleshooting, or problem determination, is the process of determining why a certain product is malfunctioning.

Note: You can resolve some problems by ensuring that your system matches the system requirements listed in Chapter 2, "Requirements and configuration for the monitoring agent," on page 5.

This appendix provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information. Also see "Support information" on page 64 for other problem-solving options.

Gathering product information for IBM Software Support

Before contacting IBM Software Support about a problem you are experiencing with this product, gather the following information that relates to the problem:

Table 6. Information to gather before contacting IBM Software Support

Information type	Description
Log files	Collect trace log files from failing systems. Most logs are located in a `logs` subdirectory on the host computer. See "Trace logging" on page 44 for lists of all trace log files and their locations. See the *IBM Tivoli Monitoring User's Guide* for general information about the IBM Tivoli Monitoring environment.
UNIX logs information	• Version number and patch level • Sample application data file (if monitoring a file) • Metafile (if problem is missing or invalid data in a workspace and the problem originates in an application that you are monitoring)
Operating system	Operating system version number and patch level
Messages	Messages and other information displayed on the screen
Version numbers for IBM Tivoli Monitoring	Version number of the following members of the monitoring environment: • IBM Tivoli Monitoring. Also provide the patch level, if available. • Monitoring Agent for UNIX Logs
Screen captures	Screen captures of incorrect output, if any.
(*UNIX only*) Core dump files	If the system stops on UNIX systems, collect core dump file from *`install_dir`*`/bin` directory, where *install_dir* is the directory path where you installed the monitoring agent.

Built-in troubleshooting features

The primary troubleshooting feature in the Monitoring Agent for UNIX Logs is logging. *Logging* refers to the text messages and trace data generated by the Monitoring Agent for UNIX Logs and is always enabled. Messages and trace data are sent to the files listed in Table 7 on page 46.

Trace data captures transient information about the current operating environment when a component or application fails to operate as designed. IBM Software Support personnel use the captured trace information to determine the source of an error or unexpected condition. See "Trace logging" on page 44 for more information.

Problem classification

The following types of problems might occur with the Monitoring Agent for UNIX Logs:

- Installation and configuration
- General usage and operation
- Display of monitoring data

This appendix provides symptom descriptions and detailed workarounds for these problems, as well as describing the logging capabilities of the monitoring agent. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Trace logging

Trace logs capture information about the operating environment when component software fails to operate as intended. The principal log type is the RAS (Reliability, Availability, and Serviceability) trace log. These logs are in the English language only. The RAS trace log mechanism is available for all components of IBM Tivoli Monitoring. Most logs are located in a `logs` subdirectory on the host computer. See the following sections to learn how to configure and use trace logging:

- "Principal trace log files" on page 45
- "Examples: using trace logs" on page 47
- "Setting RAS trace parameters" on page 48

Note: The documentation refers to the RAS facility in IBM Tivoli Monitoring as "RAS1".

Typically, IBM Software Support applies specialized knowledge to analyze trace logs to determine the source of problems. However, you can open trace logs in a text editor such as **vi** to learn some basic facts about your IBM Tivoli Monitoring environment as described in "Examples: using trace logs" on page 47.

Overview of log file management

Table 7 on page 46 provides the names, locations, and descriptions of RAS1 log files. The log file names adhere to the following naming convention:

hostname_product_program_timestamp-nn.log

where:

- *hostname* is the host name of the system on which the monitoring component is running.
- *product* is the two-character product code. For Monitoring Agent for UNIX Logs, the product code is ul.
- *program* is the name of the program being run.
- *timestamp* is an 8-character hexadecimal timestamp representing the time at which the program started.
- *nn* is a rolling log suffix. See "Examples of trace logging" on page 45 for details of log rolling.

Examples of trace logging

For example, if a UNIX logs monitoring agent is running on computer "server01", the RAS log file for the Monitoring Agent for UNIX Logs might be named as follows:

```
server01_ul_kulagent_437fc59-01.log
```

For long-running programs, the *nn* suffix is used to maintain a short history of log files for that startup of the program. For example, the kulagent program might have a series of log files as follows:

```
server01_ul_kulagent_437fc59-01.log
server01_ul_kulagent_437fc59-02.log
server01_ul_kulagent_437fc59-03.log
```

As the program runs, the first log (*nn*=01) is preserved because it contains program startup information. The remaining logs "roll." In other words, when the set of numbered logs reach a maximum size, the remaining logs are overwritten in sequence.

Each time a program is started, a new timestamp is assigned to maintain a short program history. For example, if the Monitoring Agent for UNIX Logs is started twice, it might have log files as follows:

```
server01_ul_kulagent_437fc59-01.log
server01_ul_kulagent_437fc59-02.log
server01_ul_kulagent_437fc59-03.log

server01_ul_kulagent_537fc59-01.log
server01_ul_kulagent_537fc59-02.log
server01_ul_kulagent_537fc59-03.log
```

Each program that is started has its own log file. For example, the Monitoring Agent for UNIX Logs would have agent logs in this format:

```
server01_ul_kulagent_437fc59-01.log
```

Other logs have a similar syntax as in the following example:

```
server01_ul_kulmapper_447fc59-01.log
```

where **kulmapper** is the program name.

Note: When you communicate with IBM Software Support, you must capture and send the RAS1 log that matches any problem occurrence that you report.

Principal trace log files

Table 7 on page 46 contains locations, file names, and descriptions of trace logs that can help determine the source of problems with agents.

Table 7. Trace log files for troubleshooting agents

System where log is located	File name and path	Description
On the computer that hosts the monitoring agent See "Definitions of variables" on page 47 for descriptions of the variables in the file names in column two.	The RAS1 log files are named *hostname*_ul_*program*_*timestamp*-*nn*.log and are located in the *install_dir*/logs path. **Note:** File names for RAS1 logs include a hexadecimal timestamp. **Also on UNIX, a log with a decimal timestamp is provided:** *hostname*_ul_*timestamp*.log and *hostname*_ul_*timestamp*.pid*nnnnn* in the *install_dir*/*logs* path, where *nnnnn* is the process ID number.	Traces activity of the monitoring agent. **Note:** Other RAS1 logs have a similar syntax and are located in this directory path.
	The *.LG0 file is located in the *install_dir*/logs path.	A new version of this file is generated every time the agent is restarted. IBM Tivoli Monitoring generates one backup copy of the *.LG0 file with the tag .LG1. View .LG0 to learn the following details regarding the current monitoring session: • Status of connectivity with the monitoring server. • Situations that were running. • The success or failure status of Take Action commands.
On the Tivoli Enterprise Monitoring Server See "Definitions of variables" on page 47 for descriptions of the variables in the file names in column two.	**On UNIX:** The candle_installation.log file in the *install_dir*/logs path. **On Windows:** The file in the *install_dir*\InstallITM path.	Provides details about products that are installed. **Note:** Trace logging is enabled by default. A configuration step is not required to enable this tracing.
	The Warehouse_Configuration.log file is located in the following path on Windows: *install_dir*\InstallITM.	Provides details about the configuration of data warehousing for historical reporting. **Note:** The Monitoring for UNIX Logs agent does not support historical data collection or warehousing.
	The RAS1 log file is named *hostname*_ms_*timestamp*-*nn*.log and is located in the following path: • **On Windows:** *install_dir*\logs • **On UNIX:** *install_dir*/logs **Note:** File names for RAS1 logs include a hexadecimal timestamp **Also on UNIX, a log with a decimal timestamp is provided:** *hostname*_ms_*timestamp*.log and *hostname*_ms_*timestamp*.pid*nnnnn* in the *install_dir*/*logs* path, where *nnnnn* is the process ID number.	Traces activity on the monitoring server.

Table 7. Trace log files for troubleshooting agents (continued)

System where log is located	File name and path	Description
On the Tivoli Enterprise Portal Server See "Definitions of variables" for descriptions of the variables in the file names in column two.	The RAS1 log file is named *hostname*_cq_*timestamp-nn*.log and is located in the following path: • **On Windows:** *install_dir*\logs • **On UNIX:** *install_dir*/logs **Note:** File names for RAS1 logs include a hexadecimal timestamp **Also on UNIX, a log with a decimal timestamp is provided:** *hostname*_cq_*timestamp*.log and *hostname*_cq_*timestamp*.pid*nnnnn* in the *install_dir/logs* path, where *nnnnn* is the process ID number.	Traces activity on the portal server.
	The TEPS_ODBC.log file is located in the following path on Windows: *install_dir*\InstallITM.	When you enable historical reporting, this log file traces the status of the warehouse proxy agent. **Note:** The Monitoring for UNIX Logs agent does not support historical data collection or warehousing.
Definitions of variables for RAS1 logs: • *hostname* is the host name of the system on which the agent is running. • *install_dir* represents the directory path where you installed the IBM Tivoli Monitoring component. *install_dir* can represent a path on the computer that hosts the monitoring server, the monitoring agent, or the portal server. • *product* is the two character product code. For Monitoring Agent for UNIX Logs, the product code is ul. • *program* is the name of the program being run. • *timestamp* is an eight-character hexadecimal timestamp representing the time at which the program started. • *nn* is a rolling log suffix. See "Examples of trace logging" on page 45 for details of log rolling.		

See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on the complete set of trace logs that are maintained on the monitoring server.

Examples: using trace logs

Typically IBM Software Support applies specialized knowledge to analyze trace logs to determine the source of problems. However, you can open trace logs in a text editor such as **vi** to learn some basic facts about your IBM Tivoli Monitoring environment. You can use the **ls -ltr** command to list the log files in the *install_dir*/logs directories, sorted by time they were last updated.

Example one

This excerpt shows the typical log for a failed connection between a monitoring agent and a monitoring server with the host name **server1a**:

```
(Thursday, August 11, 2005, 08:21:30-{94C}kdcl0cl.c,105,"KDCL0_ClientLookup") status=1c020006,
    "location server unavailable", ncs/KDC1_STC_SERVER_UNAVAILABLE
(Thursday, August 11, 2005, 08:21:35-{94C}kraarreg.cpp,1157,"LookupProxy") Unable to connect to
    broker at ip.pipe:: status=0, "success", ncs/KDC1_STC_OK
(Thursday, August 11, 2005, 08:21:35-{94C}kraarreg.cpp,1402,"FindProxyUsingLocalLookup") Unable
    to find running CMS on CT_CMSLIST <IP.PIPE:#server1a>
```

Example two

The following excerpts from the trace log *for the monitoring server* show the status of an agent, identified here as "Remote node." The name of the computer where the agent is running is **SERVER5B**:

```
(42C039F9.0000-6A4:kpxreqhb.cpp,649,"HeartbeatInserter") Remote node SERVER5B:KUL is ON-LINE.
. . .
(42C3079B.0000-6A4:kpxreqhb.cpp,644,"HeartbeatInserter") Remote node SERVER5B:KUL is OFF-LINE.
```

Key points regarding the preceding excerpt:

- The monitoring server appends the **KUL** product code to the server name to form a unique name (`SERVER5B:KUL`) for this instance of Monitoring Agent for UNIX Logs. This unique name enables you to distinguish multiple monitoring products that might be running on **SERVER5B**.
- The log shows when the agent started (ON-LINE) and later stopped (OFF-LINE) in the environment.
- For the sake of brevity an ellipsis (...) represents the series of trace log entries that were generated while the agent was running.
- Between the ON-LINE and OFF-LINE log entries, the agent was communicating with the monitoring server.
- The ON-LINE and OFF-LINE log entries are always available in the trace log. All trace levels that are described in "Setting RAS trace parameters" provide these entries.

Setting RAS trace parameters

Objective

Pinpoint a problem by setting detailed tracing of individual components of the monitoring agent and modules.

Background Information

Monitoring Agent for UNIX Logs uses RAS1 tracing and generates the logs described in Table 7 on page 46. The default RAS1 trace level is ERROR.

RAS1 tracing has control parameters to manage to the size and number of RAS1 logs. Use the procedure described in this section to set the parameters.

Note: The **KBB_RAS1_LOG** parameter also provides for the specification of the log file directory, log file name, and the inventory control file directory and name. Do not modify these values or log information can be lost.

Before you begin

See "Overview of log file management" on page 44 to ensure that you understand log rolling and can reference the correct log files when you managing log file generation.

After you finish

Monitor the size of the **logs** directory. Default behavior can generate a total of 45 to 60 MB for each agent that is running on a computer. For example, each database instance that you monitor could generate 45 to 60 MB of log data. See the "Procedure" section to learn how to adjust file size and numbers of log files to prevent logging activity from occupying too much disk space.

Regularly prune log files other than the RAS1 log files in the **logs** directory. Unlike the RAS1 log files which are pruned automatically, other log types can grow indefinitely, for example, the logs in Table 7 on page 46 that include a process ID number (PID).

Consider using collector trace logs (described in Table 7 on page 46) as an additional source of troubleshooting information.

Note: The maximum error tracing setting can generate a large amount of trace logging. Use them only temporarily, while you are troubleshooting problems. Otherwise, the logs can occupy excessive amounts of hard disk space.

Procedure

Specify RAS1 trace options in the *install_dir*/config/ul.ini file. You can manually edit the configuration file to set trace logging:

1. Open the trace options file: */install_dir*/config/ul.ini.
2. Edit the line that begins with **KBB_RAS1=** to set trace logging preferences.

   For example, if you want detailed trace logging, set the Maximum Tracing option:

   ```
   export KBB_RAS1='ERROR (UNIT:kul ALL) (UNIT:kra ALL)'
   ```
3. Edit the line that begins with **KBB_RAS1_LOG=** to manage the generation of log files:
   - Edit the following parameters to adjust the number of rolling log files and their size.
     - **MAXFILES**: the total number of files that are to be kept for all startups of a given program. Once this value is exceeded, the oldest log files are discarded. Default value is 9.
     - **LIMIT**: the maximum size, in megabytes (MB) of a RAS1 log file. Default value is 5.
   - IBM Software Support might guide you to modify the following parameters:
     - **COUNT**: the number of log files to keep in the rolling cycle of one program startup. Default value is 3.
     - **PRESERVE**: the number of files that are not to be reused in the rolling cycle of one program startup. Default value is 1.

   Note: The **KBB_RAS1_LOG** parameter also provides for the specification of the log file directory, log file name, and the inventory control file directory and name. Do not modify these values or log information can be lost.
4. Restart the monitoring agent so that your changes take effect.

Problems and workarounds

The following sections provide symptoms and workarounds for problems that might occur with Monitoring Agent for UNIX Logs:

- "Installation and configuration troubleshooting" on page 49
- "Troubleshooting for remote deployment" on page 59
- "Situation troubleshooting" on page 59

Note: You can resolve some problems by ensuring that your system matches the system requirements listed in Chapter 2, "Requirements and configuration for the monitoring agent," on page 5.

This appendix provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Installation and configuration troubleshooting

This section provides tables that show solutions for installation, configuration, and uninstallation problems.

Unique troubleshooting approach for Monitoring Agent for UNIX Logs

Configuration of Monitoring Agent for UNIX Logs depends on a unique configuration file (`kul_configfile`) that is located on each computer that hosts the agent software. This file specifies the following details for the monitoring of logs:

- Which log files to target for monitoring
- How to parse each line in a log
- How to map parsed strings into the database

This configuration step is unique to Monitoring Agent for UNIX Logs. Typically you specify the details of monitoring in the Situation Editor of the Tivoli Enterprise Portal, and you benefit from the features of the Situation Editor and the verification cues that you can see in the portal. These features and cues are not all available for troubleshooting Monitoring Agent for UNIX Logs. Instead you have the unique troubleshooting approaches that are described in the following sections of this document:

- Chapter 2, "Requirements and configuration for the monitoring agent," on page 5
  - "Specifying the log files to monitor" on page 8
  - "Customer configuration file" on page 8
  - "Customer configuration file format" on page 9
  - "Syslog daemon configuration file" on page 10
  - "Environment variables for the Monitoring Agent for UNIX Logs" on page 10
  - "Environment variable syntax" on page 11
  - "Dynamically refreshing the monitoring agent" on page 12
- Appendix B, "Tuning format commands with the kulmapper utility," on page 85
- Check the format strings that specify the log files, data, and fields that you monitor.
- Location defined by environment KUL_CONFIG_FILE variable described in Chapter 2, "Requirements and configuration for the monitoring agent," on page 5.
- View Log workspace in Portal. Appendix B, "Tuning format commands with the kulmapper utility," on page 85 describe how to set up the portal for this purpose.

Note: See Table 8 on page 51 to learn about a problem that affects users who have a previous version of Monitoring Agent for UNIX Logs.

Table 8. Problems and solutions for installation and configuration

Problem	Solution
When you upgrade to IBM Tivoli Monitoring, you might need to apply fixpacks to Candle, Version 350, agents.	• **Scenario 1, you upgrade monitoring agents:** Fixpacks for Candle, Version 350, are delivered as each monitoring agent is upgraded to IBM Tivoli Monitoring. **Note:** The IBM Tivoli Monitoring download image or CD provides application fixpacks for the monitoring agents that are installed from that CD (for example, the agents for operating systems such as Windows, Linux, UNIX, and i5/OS®). The upgrade software for other agents is located on the download image or CDs for that specific monitoring agent, such as the agents for database applications. • **Scenario 2, you do not upgrade monitoring agents:** If you do not upgrade the monitoring agent to IBM Tivoli Monitoring, the agent continues to work. However, you must upgrade to have all the functionality that IBM Tivoli Monitoring offers. **Note:** You might have to install fixpacks for 350 agents that you choose not to upgrade to IBM Tivoli Monitoring. Likewise, you might have to install fixpacks for any 350 agents that do not have an equivalent in IBM Tivoli Monitoring.

Table 8. Problems and solutions for installation and configuration (continued)

Problem	Solution

The Monitoring Agent for UNIX Logs agent needs pre-deploy configuration for some platforms.	The current implementation of the Monitoring Agent for UNIX Logs is not self-configuring, and can not be remotely configured. For any remote host that does not actively use syslog (for example, the default behavior on AIX and Solaris), it is necessary to pre-configure the UNIX Log Alert agent bundle in each depot that distributes them. The following steps describe the process:
	1. Ensure that the "jar" executable is available on the Tivoli Enterprise Monitoring Server. If not, a full copy of the Java SDK will need to be installed (the Java JRE typically does not provide "jar").
	2. Change the directory to the `unix` directory of the "kul" bundle that is to be updated: `.tables/`*`TEMS`*`/depot/PACKAGES/`*`ARCH`*`/kul/`*`version`*`/unix`
	For example:
	`cd $CANDLEHOME/tables/HUB_HOSTNAME/depot/PACKAGES/aix516 /kul/061000000/unix`
	3. Make a backup copy of the jar file (i.e. `ul`*`ARCH`*`.jar`)
	For example:
	`cp ulaix516.jar orig.ulaix516.jar`
	4. Unpackage the jar file: jar xf `ul`*`ARCH`*`.jar`
	For example:
	`jar xf ulaix516.jar`
	5. Edit the UNIX Log agent's configuration file: `config/ kul_configfile`. Ensure that this file *accurately* represents the log locations that are to be monitored on the destination systems.
	6. Repackage the jar file: jar cf `ul`*`ARCH`*`.jar` *`ARCH`*`/ config/ tmp/`
	For example:
	`jar cf ulaix516.jar` *`ARCH`*`/ config/ tmp/`
	7. Remove the unpackaged directories (optional): rm -rf *`ARCH`*`/ config/ tmp/`
	Extreme care should be taken during step 5, above. Once this agent has been deployed, there is no way currently available through IBM Tivoli Monitoring to update the configuration file without a login to the remote system. After the initial deployment, the `kul_configfile` can only be updated manually, directly on the remote host. Also, the agent cannot be re-deployed unless it has first been uninstalled from the remote host.

Presentation files and customized OMEGAMON® screens for Candle monitoring agents need to be upgraded to a new Linux on z/Series system.	The upgrade from version 350 to IBM Tivoli Monitoring handles export of the presentation files and the customized OMEGAMON screens.
(*UNIX only*) During a command-line installation, you choose to install a component that is already installed, and you see the following warning: `WARNING - you are about to install the SAME version of "`*`component`*`"` where *component* is the name of the component that you are attempting to install.	The system prompts you to ignore the warning and re-install the component ("Yes") or to stop installation of the component ("No"). • If you select, "Yes," you overwrite the current installation of the component. **Note:** If you had previously applied a fixpack or other modification to the component, those changes would be overwritten. • If you select, "No," you must exit and restart the installation process. You cannot return to the list where you selected components to install. When you run the installer again, do not attempt to install any component that is already installed, unless you want the installer to overwrite it.

Table 8. Problems and solutions for installation and configuration (continued)

Problem	Solution
(*Monitoring Agent for UNIX Logs only*) The *install_dir*/config/kul_configfile configuration file is empty following the installation of a new version of the Log Agent. Settings created for a previous version of this agent are lost.	This problem occurs because the installer overwrites a pre-existing copy of the kul_configfile file. Retrieve the prior version of the kul_configfile file from your archives or recreate the file from scratch. **Note:** You should rename the file or copy it to another location before upgrading to IBM Tivoli Monitoring v6.1.
The product fails to do a monitoring activity that requires read, write, or execute permissions. For example, the product might fail to read a log.	The monitoring agent must have the permissions necessary to perform requested actions. For example, if the user ID you used to log onto the system to install the monitoring agent (locally or remotely) does not have the permission to perform a monitoring operation (such as running a command), the monitoring agent is not able perform the operation.
While installing the agent from a CD, the following message is displayed and you are not able to continue the installation: `install.sh warning: unarchive of "/cdrom/unix/cienv1.tar" may have failed`	This error is caused by low disk space. Although the install.sh script indicates that it is ready to install the agent software, the script considers the size of *all* tar files, not the size of all the files that are contained within the tar file.Run the **df -k** command to check whether the file systems have enough space to install agents.
The Monitoring Agent for UNIX Logs repeatedly restarts.	You can collect data to analyze this problem as follows: 1. Access the *install_dir*/config/ul.ini file, which is described in "Setting RAS trace parameters" on page 48. 2. Add the following line: KBB_SIG1=trace –dumpoff
Agents in the monitoring environment use different communication protocols. For example, some agents have security enabled and others do not.	Configure both the monitoring server and the Warehouse proxy server to accept multiple protocols, as described in the *IBM Tivoli Monitoring Installation and Setup Guide*.
Creating a firewall partition file: The partition file enables an agent to connect to the monitoring server through a firewall.	**How it works:** When the agents start, they search KDCPARTITION.TXT for the following matches: • An entry that matches the partition name **OUTSIDE**. • An entry that also includes a valid external address. For more information, see the *IBM Tivoli Monitoring Installation and Setup Guide*.
You see the following error: `Hub not registered with location broker. Error-code 1195.`	Confirm that the password within the Tivoli Enterprise Monitoring Server is correct.
The agent is started and running but not displaying data in the Tivoli Enterprise Portal.	Perform the following steps: 1. Check the UNIX agent log files to see whether there are network connectivity problems. 2. If there are no connection problems, check whether the agent has terminated. 3. If the agent is not terminated, confirm that you have added application support for the Monitoring Agent for UNIX in the Tivoli Enterprise Monitoring Server as described in *IBM Tivoli Monitoring Installation and Setup Guide*.
The system experiences high CPU usage after you install or configure Monitoring Agent for UNIX Logs.	View the memory usage of the kulagent process. If CPU usage seems to be excessive, recycle the monitoring agent.
You see the following message: `KFWITM083W Default link is disabled for the selected object; please verify link and link anchor definitions.`	You see this message because some links do not have default workspaces. Right-click the link to access a list of workspaces to select.

Table 8. Problems and solutions for installation and configuration (continued)

Problem	Solution
When you edit the configuration for an existing monitoring agent, the values displayed are not correct.	The original configuration settings might include non-ASCII characters. These values were stored incorrectly and result in the incorrect display. Enter new values using only ASCII characters.

Table 9. General problems and solutions for uninstallation

Problem	Solution
On Windows, uninstallation of IBM Tivoli Monitoring fails to uninstall the entire environment.	Be sure that you follow the general uninstallation process described in the *IBM Tivoli Monitoring Installation and Setup Guide*: 1. Uninstall monitoring agents first, as in the following examples: • Uninstall a single monitoring agent for a specific database. —OR— • Uninstall all instances of a monitoring product, such as IBM Tivoli Monitoring for Databases. 2. Uninstall IBM Tivoli Monitoring. See the *IBM Tivoli Monitoring Troubleshooting Guide* and the section on installation problems for more information on how to remove the entire environment.
The way to remove inactive managed systems (systems whose status is OFFLINE) from the Enterprise navigation tree in the portal is not obvious.	When you want to remove a managed system from the navigation tree, complete the following steps: 1. Click **Enterprise** in the navigation tree. 2. Right-click **Workspace -> Managed System Status**. 3. Right-click the offline managed system and select **Clear offline entry**.
After uninstalling this agent, you still see "ul" agent configuration in CINFO.	The `ul.ini` file and the `ul.config` file were not deleted during the uninstallation process and still remain in the `$CANDLEHOME/config` directory. Manually delete the `ul.ini` file and the `ul.config` file from the `$CANDLEHOME/config` directory.

Table 10. General agent problems

Problem	Solution
Attributes do not allow non-ASCII input in the situation editor.	None. Any attribute that does not include "(Unicode)" might support only ASCII characters. For example "Attribute (Unicode)" will support unicode but "Attribute" without "(Unicode)" might only support ASCII characters.
The usage of the wildcard "*" when using SCAN method on a situation is not working on this agent.	The SCAN function does not support the asterisk wildcard. If used, it is treated as a literal.
The UNIX Logs agent closes.	When the agent starts, it reads the `kul_configfile` file for any logs to monitor. If there are no valid entries in the `kul_configfile` file, then it defaults to reading the `syslog.conf` to determine the logs that need to monitor. Even though the `syslog.conf` has log entries, if the entries are separated by spaces rather than tabs, the agent cannot read the `syslog.conf` file.

Table 10. General agent problems (continued)

Problem	Solution
The UNIX Logs agent might not display entries in the "Log Entries" view for a monitored log using the default query values of "current time" -12 hours.	This occurs if the portal server system is using a different local time that is behind the local time of the agent system where the UNIX Logs Agent is running. This is due to the query setting an upper bound when asking the agent to provide the contents of the monitored log which is less than the current time of the UNIX Logs system. Change the query used to populate the view with entries to set an "End Time" to a future date that accounts for the difference between portal server local time and the agent system local time. Then refresh the display in the Tivoli Enterprise Portal.

The Unix Logs agent starts up, but then stops with the following messages:
`No valid log files found in /opt/IBM/ITM/config/kul_configfile. Attempting to build default list from syslog file /etc/syslog.conf.`
`No valid log files found in /etc/syslog.conf.`
	Specify a log in the `kul_config` file to monitor. If a log is not specified, this agent has nothing to monitor and will stop.

Unable to see entries for the monitored log in the Tivoli Enterprise Portal

The UNIX Logs agent does not display entries in the Log Entries view due to the upper bound being less than the agent system time because of differences between the local times of the portal server and the agent. Define your situation to monitor the contents of the messages from the monitored logs. Ensure that you are not mixing attributes from the Universal Messages and from the UNIX Log Monitored Logs attribute groups.

For example:

```
*IF *SCAN Monitored_Logs.Log_Name *EQ 'messages'
   *AND (
          (
            ( *SCAN Universal_Messages.Message_Text *EQ 'ldap'
               *AND *SCAN Monitored_Logs.Log_Path *EQ '/var/adm/' )
        *OR ( *SCAN Universal_Messages.Message_Text *EQ 'root'
               *AND *SCAN Monitored_Logs.Log_Path *EQ '/var/adm/' )
           )
         )
```

In the example above, you are creating a situation that is looking for when an IBM Tivoli Monitoring internal message text contains "ldap" or "root", not the monitored logs contents.

The Universal Messages attribute group also contains entries like the messages you see in the `UL.LGO` file: startup and shutdown messages for situations, IBM Tivoli Monitoring status message, for example. Your situation must look like the example above, using SCAN on the Log Path, the Log Name and the attribute you are parsing from the entry in the monitored log.

Your first attribute should be one of the following from either the Monitored Logs or Log Entries attribute groups:

- Log Name The name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 128 characters.
- Log Name (Unicode) The name of the monitored log. Valid entry is a text string, with a maximum length of 384 bytes.

The value you are looking for in the situation editor should be the following:

```
abc=='/var/adm'
```

Your second attribute should be one of the following from either the Monitored Logs or Log Entries attribute groups:

- Log Path The absolute path name of the monitored log. Valid entry is an alphanumeric text string, with a maximum length of 256 characters.
- Log Path (Unicode) The absolute path name of the monitored log. Valid entry is a text string, with a maximum length of 768 bytes.

The value you are looking for in the situation editor should be the following:

```
abc=='messages'
```

For the first two attributes, you would use Monitored Log" if you want the result to be a sampled event evaluated periodically on an interval, or, if you want the result to be a pure event, you would use Log Entries.

For the third attribute, use the Log Entries attribute group, and select the appropriate attribute where the parsed message output that you want to scan is included.

From the view of the messages log, review the Log Entries view for this monitored log and pick the column that contains the data you want to scan for. For the messages log, this is probably the Description (Unicode) attribute as this contains the message ID and most of the text. In the formula for the situation, the value of the third attribute should be the following or whatever text message you looking for:

```
abc=='root'
```

Since the messages log is a system (;s) log, it is formatted differently than a user (u;) type log. You could also search the text of the other columns that are populated with data instead of the Description (Unicode) attribute. For the `/var/mqm/errors/AMQERR01.LOG` log, the third attribute would definitely be the Description (Unicode) attribute since the specified format parses the entire line of the message into the description attribute:

```
/var/mqm/errors/AMQERR01.LOG ;Y ;U ;A,"%500[^\n]"
```

So, for looking for a specific error in the `AMQERR01.LOG`, the situation formula would have Log Path (Unicode) for the first attribute (abc=='/var/mqm/errors), Log Name (Unicode) for the second attribute (abc=='AMQERR01.LOG'), and Description (Unicode) for the third attribute (abc=='AMQ9207').

Unique names for monitoring components

If you have multiple instances of a monitoring agent, you must decide how to name the monitoring agents. This name is intended to uniquely identify that monitoring agent. The agent's default name is composed of three qualifiers:

- Optional instance name

- Machine network host name
- Agent product node type

An agent name truncation problem can occur when the network domain name is included in the network host name portion of the agent name. For example, instead of just the host name myhost1 being used, the resulting host name might be myhost1.acme.north.prod.com. Inclusion of the network domain name causes the agent name in the example above to expand to SERVER1:myhost1.acme.north.prod.com:KXX. This resulting name is 39 characters long. It is truncated to 32 characters resulting in the name SERVER1:myhost1.acme.north.prod.

The agent name truncation is only a problem if there is more than one monitoring agent on the same system. In this case, the agent name truncation can result in collisions between agent products attempting to register using the same truncated name value. When truncated agent names collide on the same system, this can lead to Tivoli Enterprise Monitoring Server problems with corrupted EIB tables. The agent name collision in the Tivoli Enterprise Monitoring Server might cause a registered name to be associated with the wrong product.

In general, create names that are short but meaningful within your environment. Use the following guidelines:

- Each name must be unique. One name cannot match another monitoring agent name exactly.
- Each name must begin with an alpha character.
- Do not use blanks or special characters, including $, #, and @.
- Each name must be between 2 and 32 characters in length.
- Monitoring agent naming is case-sensitive on all operating systems.

Create the names by completing the following steps:

1. Open the configuration file for the monitoring agent, which is located in the following path:
   - **On Windows:** *&install_dir;*\tmaitm6\K*product_code*CMA.INI. For example, the product code for the Monitoring Agent for Windows OS is NT and the file name is KNTCMA.INI.
   - **On UNIX and Linux:** *install_dir*/tmaitm6/*product_code*.ini and *product_code*.config. For example, the file names for the Monitoring Agent for UNIX OS is ux.ini and ux.config.
2. Find the line the begins with **CTIRA_HOSTNAME=**.
3. Type a new name for host name that is a unique, shorter name for the host computer. The final concatenated name including the subsystem name, new host name, and UL, cannot be longer than 32 characters.

   Note: You must ensure that the resulting name is unique with respect to any existing monitoring component that was previously registered with the Tivoli Enterprise Monitoring Server.
4. Save the file.
5. Restart the agent.
6. If you do not find the files mentioned in Step 1, perform the workarounds listed in the next paragraph.

If you do not find the files mentioned in the preceding steps, perform the following workarounds:

1. Change **CTIRA_HOSTNAME** environment variable in the configuration file of the monitoring agent.
   - Find the `KULENV` file in the same path mentioned in the preceding row.
   - For z/OS® agents, find the **RKANPAR** library.
   - For i5/OS agents, find the **QAUTOTMP/KMSPARM** library in member **KBBENV**.
2. If you cannot find the **CTIRA_HOSTNAME** environment variable, you must add it to the configuration file of the monitoring agent:
   - **On Windows:** Use the **Advanced > Edit Variables** option.
   - **On UNIX and Linux:** Add the variable to the `config/`*`product_code`*`.ini` and to `config/`*`product_code`*`.config` files.
   - **On z/OS:** Add the variable to the **RKANPAR** library, member K*product_code*ENV.
   - **On i5/OS:** Add the variable to the **QAUTOTMP/KMSPARM** library in member **KBBENV**.
3. Some monitoring agents (for example, the monitoring agent for MQ Series) do not reference the **CTIRA_HOSTNAME** environment variable to generate component names. Check the documentation for the monitoring agent that you are using for information on name generation. If necessary, contact IBM Software Support.

A configured and running instance of the monitoring agent is not displayed in the Tivoli Enterprise Portal, but other instances of the monitoring agent on the same system do appear in the portal.

Tivoli Monitoring products use Remote Procedure Call (RPC) to define and control product behavior. RPC is the mechanism that allows a client process to make a subroutine call (such as GetTimeOfDay or ShutdownServer) to a server process somewhere in the network. Tivoli processes can be configured to use TCP/UDP, TCP/IP, SNA, and SSL as the desired protocol (or delivery mechanism) for RPCs.

"IP.PIPE" is the name given to Tivoli TCP/IP protocol for RPCs. The RPCs are socket-based operations that use TCP/IP ports to form socket addresses. IP.PIPE implements virtual sockets and multiplexes all virtual socket traffic across a single physical TCP/IP port (visible from the netstat command).

A Tivoli process derives the physical port for IP.PIPE communications based on the configured, well-known port for the HUB Tivoli Enterprise Monitoring Server. (This well-known port or BASE_PORT is configured using the 'PORT:' keyword on the KDC_FAMILIES / KDE_TRANSPORT environment variable and defaults to '1918'.)

The physical port allocation method is defined as (BASE_PORT + 4096*N) where N=0 for a Tivoli Enterprise Monitoring Server process and N={1, 2, ..., 15} for a non-Tivoli Enterprise Monitoring Server. Two architectural limits result as a consequence of the physical port allocation method:

- No more than one Tivoli Enterprise Monitoring Server reporting to a specific Tivoli Enterprise Monitoring Server HUB can be active on a system image.
- No more that 15 IP.PIPE processes can be active on a single system image.

A single system image can support any number of Tivoli Enterprise Monitoring Server processes (address spaces) provided that each Tivoli Enterprise Monitoring Server on that image reports to a different HUB. By definition, there is one Tivoli Enterprise Monitoring Server HUB per monitoring Enterprise, so this architecture limit has been simplified to one Tivoli Enterprise Monitoring Server per system image.

No more that 15 IP.PIPE processes or address spaces can be active on a single system image. With the first limit expressed above, this second limitation refers specifically to Tivoli Enterprise Monitoring Agent processes: no more that 15 agents per system image.

This limitation can be circumvented (at current maintenance levels, IBM Tivoli Monitoring V6.1 Fix Pack 4 and later) if the Tivoli Enterprise Monitoring Agent process is configured to use EPHEMERAL IP.PIPE. (This is IP.PIPE configured with the 'EPHEMERAL:Y' keyword in the KDC_FAMILIES / KDE_TRANSPORT environment variable). There is no limitation to the number of ephemeral IP.PIPE connections per system image. If ephemeral endpoints are used, the Warehouse Proxy Agent is accessible from the Tivoli Enterprise Monitoring Server associated with the agents using ephemeral connections either by running the Warehouse Proxy Agent on the same computer or by using the Firewall Gateway feature. (The Firewall Gateway feature relays the Warehouse Proxy Agent connection from the Tivoli Enterprise Monitoring Server computer to the Warehouse Proxy Agent computer if the Warehouse Proxy Agent cannot coexist on the same computer.)

Troubleshooting for remote deployment

Table 11 lists problems that might occur with remote deployment. This appendix provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

This section describes problems and solutions for remote deployment and removal of agent software Agent Remote Deploy:

Table 11. Remote deployment problems and solutions

Problem	Solution
While you are using the remote deployment feature to install Monitoring Agent for UNIX Logs, an empty command window is displayed on the target computer. This problem occurs when the target of remote deployment is a Windows computer. (See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on the remote deployment feature.)	Do not close or modify this window. It is part of the installation process and will be dismissed automatically.
The removal of a monitoring agent fails when you use the remote removal process in the Tivoli Enterprise Portal desktop or browser.	This problem might happen when you attempt the remote removal process immediately after you have restarted the Tivoli Enterprise Monitoring Server. You must allow time for the monitoring agent to refresh its connection with the Tivoli Enterprise Monitoring Server before you begin the remote removal process.

Situation troubleshooting

This section provides information about both general situation problems and problems with the configuration of situations. Some of the problems are generally

valid, but do not apply to Monitoring Agent for UNIX Logs. See the *IBM Tivoli Monitoring Troubleshooting Guide* for more information about troubleshooting for situations.

Specific situation problems

Table 12 lists problems that might occur with specific situations.

Table 12. Specific situation problems and solutions

Problem	Solution
You want to change the appearance of situations when they are displayed in a Workspace view.	1. Right-click an item in the Navigation tree. 2. Select **Situations** in the pop-up menu. The Situation Editor window is displayed. 3. Select the situation that you want to modify. 4. Use the **Status** pull-down menu in the lower right of the window to set the status and appearance of the Situation when it triggers. **Note:** This status setting is not related to severity settings in IBM Tivoli Enterprise Console.
When using a text editor on monitored log files you might see any of the following: • The number of events are not updated • Situations might not fire as expected • The Log Entries workspace might not show new events properly	When the monitored log files are written or modified using any text editor, the Monitoring Agent for UNIX Log's behavior over those files is undefined. Events are only expected to be updated through any application or through scripts, and not manually by opening the files using any text editor.
Situations are triggered in the Tivoli Enterprise Monitoring Server, but events for the situation are not sent to the Tivoli Enterprise Console server. The Tivoli Enterprise Monitoring Server is properly configured for event forwarding, and events for many other situations are sent to the event server.	This condition can occur when a situation is only monitoring the status of other situations. The event forwarding function requires an attribute group reference in the situation in order to determine the correct event class to use in the event. When the situation only monitors other situations, no attribute groups are defined and the event class cannot be determined. Because the event class cannot be determined, no event is sent. This is a limitation of the Tivoli Enterprise Monitoring Server event forwarding function. Situations that only monitor other situations do not send events to the event server.
Monitoring activity requires too much disk space.	Check the RAS trace logging settings that are described in “Setting RAS trace parameters” on page 48. For example, trace logs grow rapidly when you apply the **ALL** logging option.
Monitoring activity requires too many system resources.	Table 13 on page 62 describes the performance impact of specific attribute groups. If possible, decrease your use of the attribute groups that require greater system resources.
A formula that uses mathematical operators seems to be incorrect. For example, if you were monitoring Linux, a formula that calculates when **Free Memory** falls under 10 percent of **Total Memory** does not work: `LT #'Linux_VM_Stats.Total_Memory' / 10`	This formula is incorrect because situation predicates support only logical operators. Your formulas cannot have mathematical operators. **Note:** The Situation Editor provides alternatives to math operators. Regarding the example, you can select **% Memory Free** attribute and avoid the need for math operators.

Table 12. Specific situation problems and solutions (continued)

Problem	Solution
If you are running a Version 350 Monitoring Agent for UNIX Logs and you choose to alter the views to include a Version 610 UNICODE attribute, be aware that data for this attribute is not displayed and you see a blank column in this view.	To enable Unicode and other features, upgrade the monitoring agent to IBM Tivoli Monitoring, Version 6.1.0.
You see the 'Unable to get attribute name' error in the Tivoli Enterprise Monitoring Server log after creating a situation.	Install the agent's application support files on the Tivoli Enterprise Monitoring Server, using the following steps: 1. Open the Manage Tivoli Enterprise Monitoring Services window. 2. Right-click the name of the monitoring server. 3. Select **Advanced > Add TEMS Application Support** in the pop-up menu. Add application support if any for any agent that is missing from the list. See in IBM Tivoli Monitoring Installation and Setup Guide for more information on adding application support.
Events received at the Tivoli Enterprise Console server from IBM Tivoli Monitoring do not have values for all event attributes (slots) even though the values are visible in workspace views.	The problem is due to a limitation in the IBM Tivoli Monitoring interface code that generates Tivoli Enterprise Console events from situations. The situation results are provided in a chain of buffers of 3000 bytes each. The interface code currently extracts event information from only the first buffer. When situations or agent table data expands into a second buffer, this additional data is not examined, and it is not included in events sent to the Tivoli Enterprise Console server.
Tivoli Enterprise Console events from IBM Tivoli Monitoring 6.2 for IBM Tivoli Monitoring 5.x migrated situations receive parsing errors in the Tivoli Enterprise Console server.	Complete the following two steps: 1. Ensure that you have the IBM Tivoli Monitoring 6.2 Event Sync installed on your Tivoli Enterprise Console server. 2. Obtain updated baroc files from IBM Tivoli Monitoring 6.2 for the monitoring agent's events. Updated baroc files are on the Tivoli Enterprise Monitoring Server in the *CANDLEHOME*/CMS/TECLIB/itm5migr directory.
You are receiving Tivoli Business Systems Management events that cannot be associated due to application_oid and application_class not being set.	The problem is due to IBM Tivoli Monitoring 6.2 sending Tivoli Enterprise Console events for IBM Tivoli Monitoring 5.x migrated situations. These events are not able to set the cited slot values. Replace the *agent_name*_forward_tbsm_event_cb.sh script on the Tivoli Enterprise Console server with the version of this file from the Tivoli Enterprise Monitoring Server in the *CANDLEHOME*/CMS/TECLIB/itm5migr directory.
The UNIX_LAA_BAD_su_to_root_Warning situation fails.	Situation "UNIX_LAA_BAD_su_to_root_Warning" cannot be triggered on RedHat Enterprise Linux AS 5 or SUSE Linux Enterprise Server 11. It cannot be triggered on certain Linux distributions in which the file /usr/adm/suadit is not used to log failed su attempts.

Consider performance impact of each attribute group: Table 13 on page 62 lists the impact on performance (high, medium, or low) of each attribute group. The multiple-instance attributes have been classified at the lowest level. That is, the performance overhead will increase if you do not specify compare values for one or more key values.

When you want to prevent impact on performance by any of the attribute groups listed in Table 13 on page 62 you must avoid referencing that attribute group, as suggested in this list:

- Disable the attribute group.
- Never select workspaces that reference the attribute group.

- Disable situations that reference the attribute group by using the "Undistributed situations" option in the Situation Editor.
- Disable historical reporting that references the attribute group.
- Avoid using the "Auto Refresh" refresh feature in a Workspace because this option causes a refresh of data for all attribute groups.

See the *IBM Tivoli Monitoring User's Guide* for additional information on controlling attribute group usage.

Table 13. Performance Impact by attribute group

Attribute group	High	Medium	Low
Log Entries By default the table associated with the attribute group shows 24 hours of data. This set of data might be large.		✔	
Monitored Logs			✔

Problems with configuration of situations

Table 14 lists problems that might occur with situations.

This section provides information for troubleshooting for agents. Be sure to consult the *IBM Tivoli Monitoring Troubleshooting Guide* for more general troubleshooting information.

Table 14. Problems with configuring situations that you solve in the Situation Editor

Problem	Solution
Note: To get started with the solutions in this section, perform these steps: 1. Launch the Tivoli Enterprise Portal. 2. Click **Edit > Situation Editor**. 3. In the tree view, choose the agent whose situation you want to modify. 4. Choose the situation in the list. The Situation Editor view is displayed.	
The situation for a specific agent is not visible in the Tivoli Enterprise Portal.	Open the Situation Editor. Access the All managed servers view. If the situation is absent, confirm that application support for Monitoring Agent for UNIX Logs has been added to the monitoring server. If not, add application support to the server, as described in the *IBM Tivoli Monitoring Installation and Setup Guide.*
The monitoring interval is too long.	Access the Situation Editor view for the situation that you want to modify. Check the **Sampling interval** area in the **Formula** tab. Adjust the time interval as needed.
The situation did not activate at startup.	Manually recycle the situation as follows: 1. Right-click the situation and choose **Stop Situation**. 2. Right-click the situation and choose **Start Situation**. **Note:** You can permanently avoid this problem by placing a check mark in the **Run at Startup** option of the Situation Editor view for a specific situation.
The situation is not displayed.	Click the **Action** tab and check whether the situation has an automated corrective action. This action can occur directly or through a policy. The situation might be resolving so quickly that you do not see the event or the update in the graphical user interface.
An Alert event has not occurred even though the predicate has been properly specified.	Check the logs, reports, and workspaces.
A situation fires on an unexpected managed object.	Confirm that you have distributed and started the situation on the correct managed system.

Table 14. Problems with configuring situations that you solve in the Situation Editor (continued)

Problem	Solution
The product did not distribute the situation to a managed system.	Click the **Distribution** tab and check the distribution settings for the situation.
The situation does not fire. Incorrect predicates are present in the formula that defines the situation. For example, the managed object shows a state that normally triggers a monitoring event, but the situation is not true because the wrong attribute is specified in the formula.	In the **Formula** tab, analyze predicates as follows: 1. Click the *fx* icon in the upper-right corner of the Formula area. The Show formula window is displayed. a. Confirm the following details in the **Formula** area at the top of the window: • The attributes that you intend to monitor are specified in the formula. • The situations that you intend to monitor are specified in the formula. • The logical operators in the formula match your monitoring goal. • The numerical values in the formula match your monitoring goal. b. (*Optional*) Click the **Show detailed formula** check box in the lower left of the window to see the original names of attributes in the application or operating system that you are monitoring. c. Click **OK** to dismiss the Show formula window. 2. (*Optional*) In the Formula area of the **Formula** tab, temporarily assign numerical values that will immediately trigger a monitoring event. The triggering of the event confirms that other predicates in the formula are valid. **Note:** After you complete this test, you must restore the numerical values to valid levels so that you do not generate excessive monitoring data based on your temporary settings.

Table 15. Problems with configuration of situations that you solve in the Workspace area

Problem	Solution
Situation events are not displayed in the Events Console view of the workspace.	Associate the situation with a workspace. **Note:** The situation does not need to be displayed in the workspace. It is sufficient that the situation be associated with any workspace.
You do not have access to a situation.	**Note:** You must have administrator privileges to perform these steps. 1. Select **Edit** > **Administer Users** to access the Administer Users window. 2. In the Users area, select the user whose privileges you want to modify. 3. In the Permissions tab, Applications tab, and Navigator Views tab, select the permissions or privileges that correspond to the user's role. 4. Click **OK**.
A managed system seems to be offline.	1. Select Physical View and highlight the Enterprise Level of the navigator tree. 2. Select **View** > **Workspace** > **Managed System Status** to see a list of managed systems and their status. 3. If a system is offline, check network connectivity and status of the specific system or application.

Table 16. Problems with configuration of situations that you solve in the Manage Tivoli Enterprise Monitoring Services window

Problem	Solution
After an attempt to restart the agents in the Tivoli Enterprise Portal, the agents are still not running.	Check the system status and check the appropriate IBM Tivoli Monitoring logs.

Table 16. Problems with configuration of situations that you solve in the Manage Tivoli Enterprise Monitoring Services window (continued)

Problem	Solution
The Tivoli Enterprise Monitoring Server is not running.	Check the system status and check the appropriate IBM Tivoli Monitoring logs.

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online
Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.

IBM Support Assistant
The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to http://www.ibm.com/software/support/isa.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

http://www.ibm.com/software/globalization/terminology

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Documentation Central Web site at http://www.ibm.com/tivoli/documentation.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File &rarrow; Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss.

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss.
2. Select your country from the list and click **Go**.

3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at http://www.ibm.com/software/tivoli/education.

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users. Tivoli user groups include the following members and groups:

- 23,000+ members
- 144+ groups

Access the link for the Tivoli Users Group at https://community.ibm.com/community/user/imwuc/home.

Appendix A. Generic user log support

Generic User Log Support (GULS) allows you to monitor an ASCII log that does *not* conform to any of the three supported types (syslogs, errlogs, and utmp logs). This feature relies on a format command that you supply in the configuration file entry for each user log you want to monitor (see "Customer configuration file format" on page 9).

The format command describes:

- The format of the log to the monitoring agent
- How you want data that is read from the log to be mapped in the Tivoli Enterprise Portal Log Entries table view

While the data is being mapped into the table view, you have the ability to perform data type conversions (for example, decimal to hexadecimal), and do formatting to clarify the table view and facilitate the creation of situations.

Format command

A format command is composed as follows:

A, "*log format description***" ,** *data mapping [and formatting] specifications*

A The letter A.

log format description
One or more scan directives enclosed within double quotation marks. Details are provided in "Log format description" on page 69.

data mapping [and formatting] specifications
Indicates into which columns of the Log Entries table view the scanned data is mapped. Details are provided in "Data mapping specifications" on page 74.

The format description and formatting specifications both use a syntax based on that used by the standard 'C' scanf and printf functions. The format command syntax is, perhaps, best illustrated through a simple example. The format command syntax is explained in detail after the example.

Example format command

Suppose you run an application at your site that produces an ASCII log, `myapp.log`, and that you wish to monitor the messages written to this log. Suppose also that a sample entry from this log is as follows:

```
MSG123 Dec 25 2004 03:15 pm system myapp: Server frodo not responding
```

The following format command enables the monitoring agent to monitor this log allowing you to both create situations looking for specific messages and to display the log's contents within the Tivoli Enterprise Portal Log Entries table view.

```
;A , "%s %s %d %d %d:%d %s %s %[^:] : %[^\n]" , type month day year
hour minute hour system source description
```

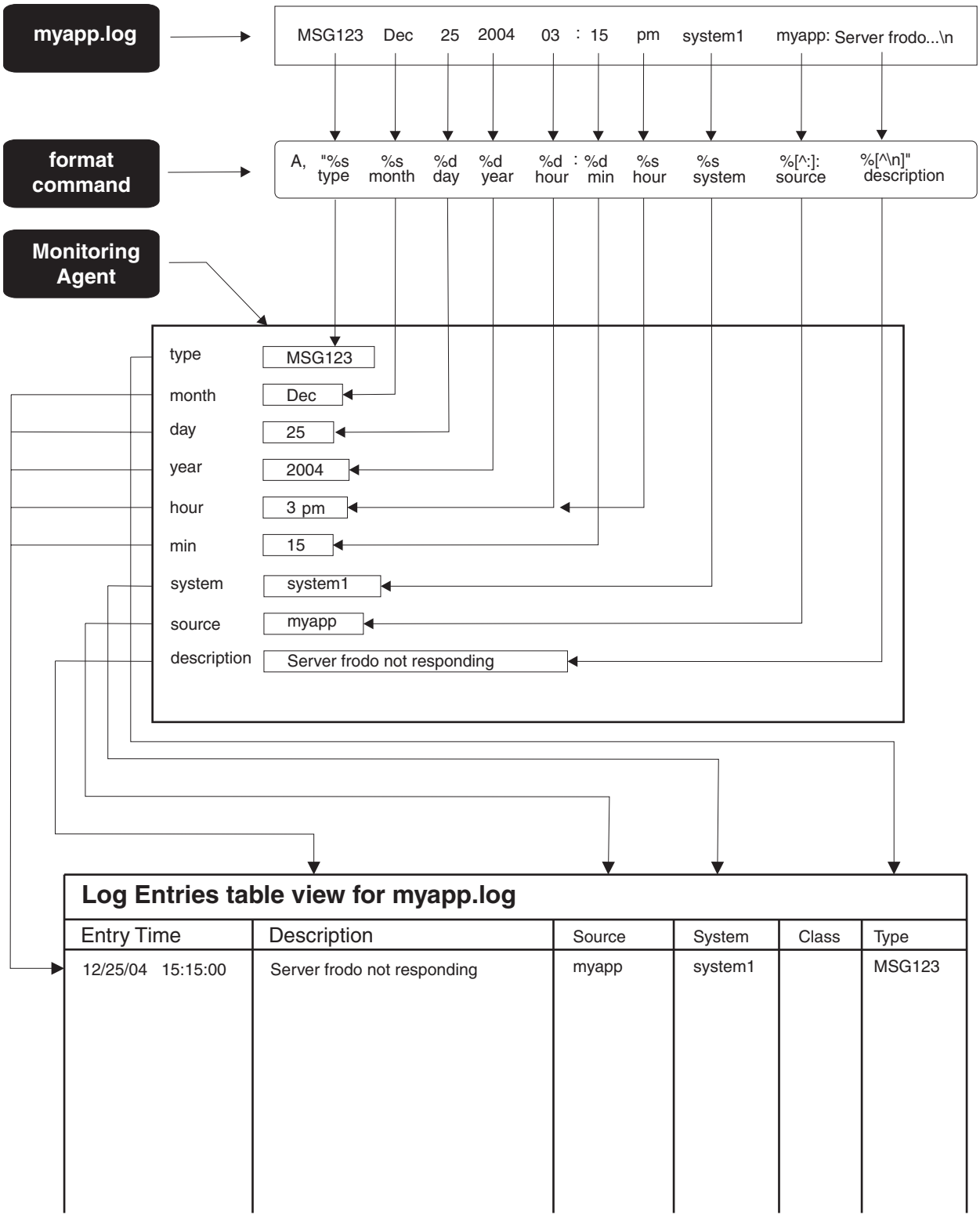

Figure 1. Example format diagram

Format command syntax

A format command consists of two components:

- The log format description
- Data mapping and formatting specifications

Log format description

The format description is comprised of one or more scan directives enclosed within double quotation marks ("..."). Generally, a scan directive identifies a field or group of fields within a log entry, although a directive can identify a single character within a log entry. A field is any sequence of nonwhite space characters terminated by one or more white space characters (that is, a tab or blank).

For example, the sequence below consists of five fields:

```
Dec 25 2004 03:15 pm
```

In addition to fields with content that varies from one log entry to another, an entry can contain fixed character strings that occur in the same relative location in all log entries. These are termed literals. Literals can be embedded anywhere within a format description.

A scan directive has the following format. (Items enclosed within brackets are optional.)

%[(offset)][*][width][size]datatype

All scan directives must include at least a percent sign, '%', and a datatype.

Each scan directive starts from where the previous one ended unless it is the first (in which case it starts at column 1 in the log entry), or an offset option has been included in the directive. Each scan directive consumes characters from a log entry until any of the following occurs:

- An inappropriate character is encountered (that is, one that does not match the expected data type).
- The field width, if specified, is exhausted.
- The end of the log entry is encountered.

Format description components

The following sections describe each of the format description components. To simplify the discussion, all examples in the next section include only the relevant scan directives from a format description. The corresponding mapping specifications that must be included in a complete format command have been omitted.

Literals: Literals describe a sequence of one or more characters that occur at the same relative location in every entry in the log, and which you do not want to map into a table view column.

Specifying a literal makes the monitoring agent look for and read those characters from a log entry, and then discard them. If you include a literal, it must match exactly the character sequence in a log entry, otherwise that entry is ignored.

For example, to read a time from a log that has the format:

```
03:15
```

The following scan directives can be used:

```
%d:%d
```

In this example, the colon ':' preceding the second directive is a literal.

Any number of white space characters that immediately precede the start of a field in a log entry are automatically consumed and discarded (unless the data type of the next scan directive is a character, for example %c). To consume any number of white space characters that are embedded within a literal in a log entry, include one or more white space characters in the format description literal. For example, suppose a log entry has the following format:

```
MSG123 < Code 9 > System1
```

If you want to extract only the message field, the code number, and the system, use the following format description:

```
%s < Code%d >%s
```

The first directive scans in the message field. The single blank following the first directive consumes all the white space between the message field and the '<' sign in the log entry. Similarly, the single blank between the '<' sign and the literal 'Code' in the format description consumes all the white space between those same literals within the log entry. No white space is required between the literal 'Code' and the '%d' directive or between the literal '>' and the '%s' directive because this white space is automatically consumed.

To include a percent sign, '%', in a literal, specify two adjacent percent sign characters in the format description. For example, if you want to extract the percentages from a log that contains the following three fields:

```
45% 82% 2%
```

Use the following format description:

```
%d%% %d%% %d%%
```

(The blanks embedded within the description are not required but clarify the example.)

Offsets: Offsets allow you to specify the absolute column within a log entry at which a scan directive starts; if no offset is specified, each succeeding scan starts where the previous one ended. The first column in an entry is 1. Offsets can facilitate the description of fixed field logs; that is, those in which a field starts in the same column in every entry.

For example, suppose each log entry starts with a message number as in the following example:

```
MSG123 Dec 25
```

If you want to extract only the message number, discarding the text "MSG" that precedes it, you can use the following scan directive:

```
%(4)d
```

This causes the scan to start in column 4, skipping over and discarding the first three characters.

Field suppression: The character '*' in a scan directive indicates that the scanned data is suppressed. That is, the data is read from the log entry but discarded. For example, suppose a log entry has the following form:

```
MSG123 Dec 25
```

If you want to skip over the message number field entirely, you can use this format description:

```
%*s %s %d
```

These directives cause the message number field to be ignored but the month and day are stored and mapped to a column. Since the data is discarded for scan directives that are suppressed, such directives have no corresponding data mapping specifier (which associates log data with a table view column).

Width: The width option allows you to specify the maximum number of characters that is consumed from a log entry to satisfy a scan directive. For example, suppose you want to extract only the first 2 digits from the message number from the following log entry:

```
MSG123 Dec 25
```

You can use the following scan directives:

```
%*3s %2d %*d
```

The first directive, "%*3s", discards the first three characters of the message number field (the text "MSG"). The second directive, "%2d", saves the digits "12" for mapping and the last directive, "%*d", discards any digits that remain in the message field. The digit '3' is consumed and discarded.

For scan directives that have a data type of "string" (that is, %s or %[]), the default width is 31.

Size: The size option can be used in numeric (that is, non-string) directives and controls the amount of storage that is reserved to hold a scanned number. Allocating more storage allows larger numbers to be scanned and stored. Unless you need to scan very large numbers or need to increase the precision, the default sizes are probably sufficient. The effect of including a size option in a numeric scan directive depends on the operating system on which the monitoring agent is run.

The valid size option and data type combinations that you can specify are listed in the following table.

Table 17. Monitoring Agent for UNIX Logs valid size option and data type combinations

Size option	Can be used with these data types
l	d, i, o, u, x, e, f, g
ll	d, i, o, u, x
L	e, f, g
h	d, i, o, u, x

If you explicitly include formatting in the data mapping specifier for a scanned value rather than allowing the print directive to default, the size option that you specify in the scan and print directives must be consistent.

Data type: The data type of a scan directive indicates whether the corresponding characters in the log entry are alphanumeric or numeric and affects how the data is stored by the monitoring agent once it has been scanned. Specifying that log data is numeric instead of a simple alphanumeric string can simplify the format description and allows the scanned data to be converted (for instance, displayed in hexadecimal or scientific notation instead of decimal), as it is being mapped into a table view column.

The following table lists the valid data types you can use in a scan directive to describe alphanumeric data.

Table 18. Monitoring Agent for UNIX Logs valid alphanumeric data types

Data type	Corresponding field in the log entry
s	A sequence of nonwhite space characters. Characters from the log entry are consumed until the first white space character is encountered or until the number of characters specified in the field width has been exhausted. If no width is specified in the directive, the default width is 31.

c	A sequence of bytes. The number of bytes consumed is determined by the specified width option. If no width is specified, the default is 1. Unlike all other data type directives, white space immediately preceding the corresponding field in the log entry is not automatically skipped. To skip over white space, you must explicitly include a white space literal immediately preceding a directive with a data type of character.
	
	This feature is useful for describing fields in a log entry that might be blank assuming the starting column and width of the optional field is known. For example, suppose two entries from a log are as shown:
	`field1a field2a field3a`
	`field1b field3b`
	
	Field 2, if present, always starts in column 12 and can be up to 9 characters long. The following format description could be used:
	`%s %(12)9c %s`
	
	In this example, the second directive will store “field2a” when the first entry is processed and will contain blank when the second entry is processed.

Table 18. Monitoring Agent for UNIX Logs valid alphanumeric data types (continued)

Data type	Corresponding field in the log entry

[inclusive scanset] or [^exclusive scanset	Any sequence of characters. A scanset data type is a generalized type 's' (string) data type. In fact, the type 's' directive can be expressed by the following exclusive scanset:
	`%[^ \t\n]`
	
	This says that all non-blank, non-tab and non-newline characters will be consumed. That is, the scan will end on the first white space character in the log entry. (See "Escape characters" on page 79 for details on specifying escape characters in a format command.)
	
	In an inclusive scanset, characters from a log entry will be consumed until a character is encountered that is not in the scanset. In an exclusive scanset (for example, one that has a '^' (circumflex) character immediately following the left bracket), characters from a log entry will be consumed until a character is encountered that is in the scanset.
	
	A scanset allows a single scan directive to consume multiple log entry fields. For example, a scanset that you might use frequently is one that is to "read all the remaining characters in an entry":
	`%[^\n]`
	
	That is, consume everything from the current position in an entry up to the newline character, which marks the end of the entry.
	
	A scanset directive can also be used to terminate a scan before a simple type 's' variable would, that is, when a white space character is found. For example, suppose a log entry has embedded within it either one of the following two field sequences:
	`Error code:24`
	`Warning code:16`
	
	If you want only to extract the numeric code itself, the following directives could be used:
	`%*[^:]:%d`
	
	This format description consumes and discards (field is suppressed) all characters until a colon is found (exclusive scanset), then consumes and discards the colon itself (an embedded literal) and finally consumes and stores the numeric code.
	
	As with a string data type, if you wish to consume more than 31 characters with a scanset directive, you must include a maximum width option in the directive. For example, to consume up to 60 characters from the current location up to the end of the entry use:
	`%60[^\n]`
	
	Some operating systems support the use of a '-' (dash) to represent a range of characters, for example:
	`%[a-z]`
	
	This example includes all lowercase letters in the scanset. The character that precedes the dash must be lexically less than the character following it otherwise the dash stands for itself. Also, the dash stands for itself whenever it is the first or last character in the scanset.
	
	To include the right bracket in an inclusive scanset, it must immediately follow the opening left bracket. To include the right bracket in an exclusive scanset, it must immediately follow the circumflex character. In both cases, a right bracket so placed is not considered the closing right bracket of the scanset.

Table 18. Monitoring Agent for UNIX Logs valid alphanumeric data types (continued)

Data type	Corresponding field in the log entry
d, u	An optionally signed decimal integer.
i	An optionally signed integer with a base determined by the first characters of the number: • If the first character is in the range 1 to 9, the base is 10. • If the first character is 0 and the second character is in the range 0 to 7, the base is 8. • If the first 2 characters are 0x (or 0X), the base is assumed to be 16; that is, all characters in the range 0 to 9 and a to f (or A to F) are considered part of the number.
o	An optionally signed octal integer, that is, a string of integers in the range 0 to 7.
x	An optionally signed hexadecimal integer, that is, a string of characters in the range 0 to 9 and a to f or A to F.
e,f,g	An optionally signed string of digits that can contain a decimal point and/or an exponent component that consists of an 'E' or an 'e' followed by an optionally signed integer.

Data mapping specifications: The data mapping specifications that comprise the second component of a format command are separated from the format description by a single comma. Each mapping specification is separated from the next by white space. Every non-suppressed scan directive in the format description must have a single, corresponding data mapping specifier to indicate into which column of the Log Entries table view the scanned data must be mapped. That is, the general form of a format command is as follows:

```
A , "%scan1 %scan2 %*scan3 %scan4" , mapspec1 mapspec2
mapspec4
```

The third scan directive has no corresponding mapping specifier since it is suppressed (*).

As the data is mapped into a table view column you can also optionally specify how you want it formatted and (for data read in and stored in numeric form), that numeric data is converted to a different type (for example, decimal to hexadecimal or exponent form). See "Specifying log entry times" on page 81 for further details.

The columns into which scanned data can be mapped correspond to columns in the Log Entries table view. The following table lists all the valid column names and minimum abbreviations that can be used.

Table 19. Log entries table view column mapping names

Tivoli Enterprise Portal Log Entries table view column name	Format command mapping name	Minimum abbreviation
Entry Time	month	mo
	day	da
	year	ye
	hour	ho
	minute	mi
	second	se
Description	description	de

Table 19. Log entries table view column mapping names (continued)

Tivoli Enterprise Portal Log Entries table view column name	Format command mapping name	Minimum abbreviation
Source	source	so
System	system	sy
Class	class	cl
Type	type	ty

Referring again to the example at the beginning of this section ("Example format command" on page 67), notice that there are 10 scan directives and also that there are 10 mapping specifications. Each successive scan directive, reading the format description left to right, is associated with each successive mapping specifier. That is, the first log entry field, MSG123, is read by the first scan directive, %s, and is mapped to the first column specifier, type. The second field, Dec, is read by the second scan directive, %s, and is mapped to the second column specifier, month, and so on.

As indicated in this same example, it is possible to concatenate two or more non-adjacent log entry fields into the same table view column. The log entry time is in 12-hour format; that is:

```
03:15 pm
```

The corresponding format description and data mapping specifiers in the example format command are:

```
"%d:%d %s" , hour minute hour
```

This causes '03' to be read by the first scan directive, '%d', and mapped into the hour column. The next character in the log entry, the colon, matches the colon literal in the format description and is discarded. The '15' is read by the second scan directive, '%d', and is mapped into the minute column. The 'pm' is read by the third scan directive, '%s', and is also mapped into the hour column. The result is that the hour and minute columns contain '3pm' and '15' respectively. (If the monitoring agent is passed an hour of '3pm' it converts this into 24-hour format for display in the Log Entries table view. See "12-Hour format times" on page 81 for more information concerning valid date and time formats that can be passed to the monitoring agent.)

The example above shows that it is necessary only to supply a column name into which scanned log data is mapped. For each mapping specification that has no explicit format specifier, default formatting is applied. The monitoring agent expands the mapping specifiers in the previous example as follows:

```
"%d:%d %s" , hour="%d" minute="%d" hour="%s"
```

How to override the default mapping format specifiers is the subject of the next section.

Formatting mapped data: If no formatting is included for a given mapping specification, a default format specification is assigned based on the data type and data type size in the corresponding scan directive. To specify explicitly how scanned data is formatted as they are mapped into a given column, follow the column map name with an equal sign (=) and enclose your format specifier in double quotation marks ("...").

The syntax for a data mapping specification that includes a format specifier is as follows:

MappingName[=“[literals]%[options][width]
[.precision][size]datatype[literals]”]

The MappingName corresponds to one of the full or abbreviated names from columns 2 and 3 of Table 19 on page 74 and items in brackets are optional. Since there is a one-to-one relationship between a scan directive in the format description and a mapping specification, include at most one “%datatype” directive in a mapping format specification.

Mapping format specifier components

The following paragraphs describe each of the mapping format specifier components.

Literals: Literals can be included before the scanned data is mapped into a table view column, afterwards, or both, and can serve to clarify the Log Entries table view and facilitate the creation of situations. For example, suppose you are monitoring a log that includes the following three fields:

```
... 13303 15 4 ...
```

The first field indicates a process identifier, the second represents a return code and the third a severity. You might choose to map and format the data as follows:

```
"... %d %d %d ..." , ... source="proc id. = %d" desc="RC = %d " desc=";
Severity = %d"...
```

(The ellipses represent omitted fields.) This causes the following to be displayed in the source and description columns in the Log Entries table view:

Source	Description
proc id. = 13303	RC = 15 : Severity = 4

To include a single % (percent sign) in a column, include two consecutive % characters in the format specifier. For example, desc=“%d%%” maps the integer 83 as “83%” into the description column.

Options: One or more options can be included in a mapping format specifier although not all combinations are valid. The options and their meaning are shown in the following table.

Table 20. Log Entries table view mapping options

Option	Description	Notes
‘	Formats integer portions for i, d, u, f, g and G data types with the appropriate thousands grouping separator.	Effect depends on the locale setting on the managed system. Use the locale -a command to view available locales and review the monitoring agent log to determine the current locale.
-	Left-justifies data if number of characters mapped is less than the minimum field width (see below).	Use with field width.

Table 20. Log Entries table view mapping options (continued)

Option	Description	Notes
+	Inserts a '+' or '-' sign before a numeric value depending on whether it is greater or less than zero.	Valid only for signed numeric data types.
space character	Inserts a space character before a positive numeric value; inserts a '-' sign before a negative value.	Valid only for signed numeric data types. Ignored if '+' option is also specified.
#	For 'o' data type, increases precision to force the first digit of the result to be a zero. For 'x' and 'X' data types, precedes a nonzero result with '0x' or '0X' respectively. For 'e', 'E', 'f', 'g' and 'G' data types, the result always contains a decimal point even if no digits follow it. For 'g' and 'G' data types, trailing zeros are not removed from the result.	Not valid for c, d, s or u data types.
0	Pads to the field width with leading zeros.	Valid only for numeric data types. Ignored if '-' option is also specified. Ignored also for d, i, o, u, x and X data types if precision is specified.

For example, suppose a log entry contains the character sequence "65000" and the format command contains:

```
"... %d ..." , ... type="%'0+9d"
```

The type column is displayed as:

```
+0065,000
```

Width: A decimal digit string included in a mapping format specifier signifies the minimum width of the field into which the data is mapped. If the mapped data contains fewer characters than the minimum field width, it is right justified and padded on the left to the length specified by the field width. If the '-' (left-justify) option has been specified, the data is padded on the right.

For example, suppose the log entry contains the following fields:

```
1 789 82 4567
```

You supplied the following format command:

```
A, "%d%d%d%d" , desc="%8d" desc="%8d" desc="%8d" desc="%8d"
```

The description column is formatted as follows:

```
1 789 82 4567
```

A field width with a leading zero is interpreted as meaning that the field must be 0 padded.

Precision: A precision is specified by a [.] (dot) followed by a decimal digit string. The effect of a precision depends on the type of data being mapped.

Table 21. Mapping precision and the data types specified

Data type	Precision specifies
d, i, o, u, x, X	The minimum number of digits to appear
e, E, f	The number of digits to appear after the decimal point
g, G	The maximum number of significant digits
strings	The maximum number of bytes to be printed from a string

For example, suppose a log entry contains the following fields:

```
2 3.142857 123.45 On no account allow a vogon to read poetry at you
```

The format command is:

```
A,"%d%f%g%[^\n]" , de="%.3d" de=" %.3f" de=" %.2g" de=" %.13s"
```

The description field contains the following:

```
002 3.143 1.2e+02 On no account
```

Size: The valid size options that you can specify in a format specification for mapped data are the same as those that can be specified in a scan directive in the log format description. See Table 17 on page 71 for a list of valid size and data type combinations.

The data size specified in a mapping format specifier must be the same as that specified in the corresponding scan directive.

Data type: As with the size option, the data type in a mapping format specifier must be consistent with that in the corresponding scan directive. This means that if data is scanned and stored as an integer, it must be mapped as an integer. If it is scanned as a floating point number, it must be mapped as a floating point number. If it is scanned as a character or character string it must be mapped as a character or character string.

Numeric data can be scanned and stored in one of two families: the integer family and the floating point family. Within each family, data can be represented in different ways. An integer can be displayed in decimal, octal, hexadecimal and unsigned formats. A floating point number can be displayed in decimal or exponent notation. When numeric data is mapped, it is legal to use a different data type to that in the scan directive as long as the format data type comes from the same family. Put another way, it is not legal to mix data types from different numeric families.

This feature allows you to perform type conversions as you map data into a table view column to clarify the table view. For example, if a field in a log entry contains the size of a file in bytes, you can display the size as a hexadecimal value in the Log Entries table view by using the following in the format command:

```
A, "... %d ..." , ... desc = "File size is %#x bytes" ...
```

If the file size is, for example, 11259375 bytes, the description column will contain:

```
File size is 0xabcdef bytes
```

For an example that mixes data types from the floating point family, the size can be displayed in exponent notation with the following format command:

```
A, "... %f ..." , ... desc = "File size is %.2e bytes" ...
```

This time, the description column would contain:

```
File size is 1.13e+07 bytes
```

The valid mapping data types are specified by family in the following tables.

Table 22. Integer family data types

Data type	Format of data scanned and stored as integers
d, i	Signed decimal numbers.
u	Unsigned decimal numbers.
o	Unsigned octal numbers.
x, X	Unsigned hexadecimal numbers. The letters abcdef are used for x; the letters ABCDEF are used for X.

Table 23. Floating point family data types

Data type	Format of data scanned and stored as integers
f	Signed decimal numbers with the number of digits after the decimal point equal to the precision. If no precision is specified, the default is 6 digits.
e, E	In exponent form, that is, [-].ddd+/-dd. One digit precedes the decimal point and the precision specifies the number of digits that follow it. The default precision is 6. The E data type produces a number with E instead of e before the exponent.
g, G	In either the f or g (G if E is used) formats, depending on the value of the number with the precision specifying the number of significant digits. The exponent form is used if the exponent is less than -4 or greater than the precision.
c	As a single character. The mapped data can have been read as a single character, '%c', or could be the first character of a string that was stored using a scan directive such as '%s', '%[]', or '%nc' (where n is an integer field width).
s	As a string. The mapped data must have been read and stored as a string using a scan directive such as '%s', '%[]', or %nc' (where n is an integer field width greater than 1). All characters from the string are printed up to the number of bytes indicated by the precision.

Escape characters: You might want to include characters in a format command that either are not valid in the command itself (for example, the newline character), or which have a special meaning to the monitoring agent when it is interpreting the format command (for example, the double quote, ", character). Such characters are represented in the format command by a sequence of two characters:

- The backward slash (\) escape character
- Following the backward slash, a character that represents the character code that is being escaped

The backward slash character removes any special meaning from the following character and causes the latter's single character code value to be substituted instead.

Escape characters can be included in three areas of a format command:

- In the format description
  - As part of a literal

  - Inside a scanset (%[]) type scan directive
- In a mapping specifier
  - As part of a literal

For example, suppose you want to monitor a log that contains the following sample entry:

```
"http://www.acme.com" : GET /download/Acme.exe
```

Suppose that you want to extract the character string between the set of double quotation marks and everything after the colon. Also, assume you want to map the first string into the source column of the Log Entries table view and that you want to map the second string into the description column enclosing it in quotation marks. The following format command accomplishes this:

```
A,"\"%[^\"]\" :%[^\n]" , source desc = "\"%s\""
```

Following the double quotation mark that starts the format description is an escaped double quotation mark literal that consumes the double quotation mark preceding "http:" in the sample log entry. The exclusive scanset scan directive contains an escaped double quotation mark that terminates the first scan; that is, "http://www.acme.com" is stored by the scanset directive. The escaped double quotation mark, white space character, and colon literal following the scanset directive consumes all characters up to the text "GET" in the sample log entry. The second scanset consumes and stores all characters until a newline character is encountered (end of line). The next non-escaped double quotation mark terminates the format description component of the format command.

The mapping specification for the description column contains two escaped double quotation mark literals surrounding the scanned string.

The following shows how the sample data above is displayed in the Log Entries table view using this format command.

Description	Source
"GET /download/Acme.exe"	http://www.acme.com

The following table lists all the characters that can be represented by an escape sequence and the associated escape sequence.

Table 24. Escape character sequence

Character	Escape sequence representation
newline	\n
horizontal tab	\t
vertical tab	\v
backspace	\b
carriage return	\r
backspace character (\)	\\
single quotation mark (')	\'
double quotation mark (")	\"
alert	\a

Specifying log entry times

The entry time displayed in the Log Entries table view is extracted from each log entry and is not the time at which the log monitor detected the event. The format command that you supply for each user-defined log must, therefore, include scanning and mapping specifications for the entry time.

Since the format of dates and times varies so widely between logs, the Entry Time column of the Log Entries table view is composed of six components: the year, month, day, hour, minute, and second. You specify scanning and mapping pairs explicitly for each component using one of the mapping names in Table 19 on page 74.

When the monitoring agent formats an entry from a log, it attempts to build a timestamp with a format of:

```
mm/dd/yy hh:mm:ss
```

'yy' is the 2 digit year, the first 'mm' is the month, 'dd' is the day of the month, 'hh' is the hour (in 24-hour format), the second 'mm' is the minute and 'ss' is the second. To do so, the monitoring agent expects that the data that is mapped into each of the entry time component fields is a valid integer. The data type with which each component was scanned and mapped is not important; what is important is that the formatted result is an integer.

For example, suppose the date and time in a log entry is in the form:

```
MSG123 2005 03 03 10 15 56 ...
```

Use the following format command to extract and map the entry time components:

```
A , "%s %s %s %s %s %s %s" , desc year month day hour minute
second
```

The following format command is also valid:

```
A , "%s %d %d %d %d %d %d" , desc year month day hour minute
second
```

There are two exceptions to the requirement that all time components consist of numeric data only: text months and 12-hour format times.

Text months: If the month of the log entry is in text form, for example, Jan, Feb or JAN, FEB, you can read and map the month as a string. When the monitoring agent is passed a month in this format, it will translate it to the appropriate numeric value when constructing the entry time.

12-Hour format times: Some logs contain the entry time in 12-hour format, for example, 03:15 pm. Since the monitoring agent displays entry times in 24-hour format, for such logs you must pass the 'am/pm' indicator to the monitoring agent in the hour column. An example format command for reading and mapping the hour and minute from a log with this format follows.

```
"%d:%d%s" , hour minute hour
```

This concatenates the 'am/pm' indicator to the 12-hour value so that, using the previous time as an example, the value "3pm" is passed to the monitoring agent. When constructing the entry time for an event, the monitoring agent will translate such a time to its equivalent 24-hour format, in this case '15'.

Hardcoding missing entry time components

If you omit a scan or map specification pair for the seconds component of the entry time, it will default to zero. If you omit a scan or map pair for any of the other entry time components, the monitoring agent will default the value to the corresponding current value reported by the system clock both for real-time, monitored events and for events formatted for a table view request (see the exception which follows for table view requests if the omitted value is the year). This can be appropriate for the purposes of monitoring but can lead to unpredictable or misleading results for table view requests (which return all entries from a given log that occurred within a certain time span).

If, for instance, the minute is not supplied within a log entry and you do not want to let the minute default to the current system clock minute for either monitored events or table view requests, you can hardcode a value such as '0' for the minute column. Suppose an entry from such a log has the following format:

```
MSG123 2005 Mar 6 10 pm Text of event ...
```

The following format command will hardcode a value of zero for the entry time minute for every event:

```
A,"%s %d %s %d %d %s%c %[^\n]" , de ye mo da ho ho min="0"
desc=":%s"
```

The Tivoli Enterprise Portal Log Entries table view would contain the following in the Entry Time and Description columns. (If omitted, seconds defaults to zero.)

Entry Time	Description
03/06/05 22:00:00	Text of event

To include a mapping specifier for the 'minute' column, there must be an associated scan directive. However, we are trying to hardcode a value of zero for this column; the data consumed by the associated scan directive will not be used and is, therefore, totally arbitrary.

In this example, a dummy scan directive is supplied '%c' that consumes the single space character between the 'pm' and the start of the actual message. Since this space was going to be discarded anyway and does not affect the data in interest, this scan directive's sole purpose is to allow the inclusion of a 'minute' column mapping specifier in which a '0' character is forced to be displayed.

Defaulting the year entry-time component

Many logs omit the year from their event entry times. For example, the following sample was taken from a syslog.

```
Mar 17 03:34:11 frodo unix: NFS server gandalf not responding
```

When monitoring such logs, the monitoring agent sets the entry-time year component for each new event to the current system-clock year as described in "Hardcoding missing entry time components."

When handling table view requests for logs that do not include the entry-time year, the monitoring agent attempts to determine the year of an event based on the date in the next entry. This causes the monitoring agent to make an assumption that a monitored log has never been inactive for a period one year or longer. To show how this works, suppose two entries from a syslog are as follows. (New log entries are appended to the end of the log so the entry dated December 31st is older than that dated January 1st.)

```
Dec 31 23:34:11 bilbo unix: NFS server gandalf not responding
Jan 1 03:34:11 frodo unix: NFS write error on host bilbo
```

Further, suppose that the current date is March 15th, 2005. If you issued a table view request and specified in the time span dialog a time range of December 31st, 2004 at 11:00 p.m. to January 1st, 2005 at 4:00 a.m., the above entries are displayed (in reverse chronological order) in the Log Entries table view as follows:

Entry Time	**Description**
01/01/05 03:34:11	NFS server gandalf not responding
12/31/04 23:34:11	NFS write error on host bilbo

Appendix B. Tuning format commands with the kulmapper utility

You can use the kulmapper utility to create and fine-tune format commands.

Running the utility invokes the same code as that used by the Monitoring Agent for UNIX Logs to monitor user logs. This means that a format command that maps a user log as desired when passed to kulmapper will also work correctly when used by the Monitoring Agent for UNIX Logs to monitor that log. Conversely, if a format command is passed to the utility to format a given user log, any errors in the format command will produce exactly the same messages as those generated by the monitoring agent if it were given the same format command to monitor the same log.

It is easier to build and test format commands using the utility instead of the Monitoring Agent for UNIX Logs for several reasons:

- The format command used by kulmapper is included in the same file that the utility reads that contains your sample user log entries. This is of benefit not only because it is easier to create a format command while you can see actual log entries but also because it simplifies the task of managing multiple sample user log files and their corresponding format commands.
- Using sample user log files means you can edit the log entries and play "what if" scenarios to ensure your format command can handle the different messages that can be written to the log.
- All messages written by kulmapper are sent to the standard output device, which, by default, is your terminal. The monitoring agent, on the other hand, sends any syntax and formatting error messages to its RAS log. If formatting is successful, the results must be viewed in the Log Entries table view of Tivoli Enterprise Portal.
- The default kulmapper RAS tracing options cause each line read from the sample log file, including the format command itself, to be displayed followed by the results of formatting and mapping that entry. This simplifies the task of verifying that each sample log entry was formatted as expected. If an entry is encountered that cannot be formatted, kulmapper stops after displaying the log entry that caused the error and the error message describing the problem.
- After you have updated the format command in the kulmapper sample log file, you test it by simply invoking the utility again and observing the results on your terminal. Conversely, after updating a format command in the monitoring agent's configuration file, it is necessary to send the monitoring agent a refresh signal so that it can learn and use the new format.

Using the kulmapper utility you can significantly reduce the time required performing each "change-test-observe" cycle as you tune format commands.

Using the kulmapper utility

The kulmapper utility is invoked using a script called kulmapper that is located in the *install_dir*/bin directory. The script accepts three parameters as follows:

kulmapper [*install_dir***] [***filename***]**

All parameters are optional.

-h *install_dir*
The name of the top-level directory in which you installed the monitoring agent.

-l *log_filename*
The name of a file that contains a format command and one or more sample entries from your user log.

If unspecified, the input file defaults to *install_dir*/config/kulmapper.samp and the number of entries to format defaults to 1. Here is a sample file that can be used by kulmapper:

```
a,"%9s%c/%d %s %d/%d %d:%d:%d %s %s :%[^\n]" , desc type class =
"RC = %x" month day year hour min sec hour source desc = " %s"
MSG123456I/1024 Oct 04/05 11:15:32 am region1 : Application alpha
started
MSG234567W/2048 Oct 04/05 1:01:31 pm region2 : No journal files
opened
MSG345678E/4096 Oct 04/05 2:57:02 pm region1 : Unable to open file
'FILE1'
```

To see how the utility operates, simply change to the *install_dir*/bin directory and enter the script name, kulmapper, at the command prompt without any parameters. This formats and maps the first sample log entry in the file according to the supplied format command.

For example, if your installation directory is /myinstall_dir, enter the following:

```
cd /myinstall_dir
kulmapper
a,"%9s%c/%d %s %d/%d %d:%d:%d %s %s :%[^\n]" , desc type class =
"RC = %x" month day year hour min sec hour source desc = " %s"
MSG123456I/1024 Oct 04/05 11:15:32 am region1: Application alpha
started
year: 05
month: Oct
day: 4
hour: 11am
minute: 15
second: 32
system:
source: region1
type: I
class: RC = 400
```

The contents of *install_dir*/config/kulmapper.samp is based on “Monitoring user and third-party vendor applications” on page 17 and indicates the required format of the kulmapper input file.

Analyzing User log files and testing format commands

Before using kulagent or kulmapper, analyze the log file that you want to monitor to determine how you want the data to be mapped to the Monitoring Agent for UNIX Logs tables. The name of each attribute in the Monitoring Agent for UNIX Logs tables is documented in Chapter 4, "Attributes reference," on page 21. When you have completed your analysis, write a format command, create a test log file with the format command as the first record, and use kulmapper to test the command.

For example, perform the following steps:

1. Locate a sample of the log file that you want to monitor and analyze each field of the log file.

   For example, your log file contains a message like the one following:

   ```
   ERZ010117I/0150 11/01/05 10:20:51 xcicsol1 : CICS is performing a cold
   start
   ```

2. Determine which field you want mapped to each mapping name.

   The Monitoring Agent for UNIX Logs mapping names are: year, month, day, hour, minute, second, type, system, source, class and description. In the log file record example, map the date/timestamp to the date/time mapping names. Next, let's map the message number ERZ010117I/0150 to the type mapping name. Next, map the CICS® name to the system mapping name, and the remainder of the message to the description mapping name. Finally, the remaining mapping names, source and class are not used, so the monitoring agent will assign them blank values.

3. Write your data mapping specification.

   From our analysis of the example, the mapping specification is:

   ```
   Type month day year hour minute second system description
   ```

   This is the order in which this data is displayed in the log record, so specify them in that order.

4. Determine the character type of each field.

   Basically, the character type will be a character string or a numeric string, and you need to identify which type applies to each mapping name. For example, a date can be 01/01/05 or 01 January, 2005, so the month can be numeric or character. In the example, the following are included:

   - ERZ010117I/0150 = character string
   - 11/01/05 = numeric/numeric/numeric. Ignore the "/" character when the data is mapped.
   - 10:20:51 = numeric:numeric:numeric. Ignore the ":" character.
   - xcicsol1 = character string
   - : Ignore the colon that follows the system name.
   - Keep the remainder of the message.

5. Write the format description.

   Using the example, the format description is:

   ```
   "%s %d/%d/%d %d:%d:%d %s : %[^\n]"
   ```

   Note that the literal characters "/" and ":" are included in the format description. This tells the Monitoring Agent for UNIX Logs to skip over them and omit them from the mapping variable's value. The format scan set "%[^\n]" specifies that the data is character data and to include all of it up to the new line character at the end of the log record.

6. Combine the mapping specification and the format description into a format command.

   The final format command for the example looks like this:

   ```
   a,"%s %d/%d/%d %d:%d:%d %s : %[^\n]", type month day year hour
   minute second system descr
   ```

7. Build the kulmapper test file and test.

   All that remains to test the format command is to copy it as the first record of the log file that you are analyzing, and test it with kulmapper.

Appendix C. IBM Tivoli Enterprise Console event mapping

Each event class corresponds to an attribute group in the monitoring agent. For a description of the event slots for each event class, see Table 25 on page 90. For more information about mapping attribute groups to event classes, see the *IBM Tivoli Monitoring Administrator's Guide*.

Generic event mapping provides useful event class and attribute information for situations that do not have specific event mapping defined. BAROC files are found on the Tivoli Enterprise Monitoring Server in the installation directory in TECLIB (that is, *install_dir*/cms/TECLIB for Windows systems and *install_dir*/tables/*TEMS_hostname*/TECLIB for UNIX systems). For information on the current version of the BAROC file, see the *IBM Tivoli Monitoring Installation and Setup Guide*. IBM Tivoli Enterprise Console event synchronization provides a collection of ready-to-use rule sets that you can deploy with minimal configuration. Be sure to install IBM Tivoli Enterprise Console event synchronization to access the correct Sentry.baroc, which is automatically included during base configuration of IBM Tivoli Enterprise Console rules if you indicate that you want to use an existing rulebase. See the *IBM Tivoli Monitoring Installation and Setup Guide* for details.

To determine what event class is sent when a given situation is triggered, look at the first referenced attribute group in the situation predicate. The event class that is associated with that attribute group is the one that is sent. This is true for both pre-packaged situations and user-defined situations. See the table below for attribute group to event classes and slots mapping information.

For example, if the situation is monitoring the Monitored Events attribute from the Monitored_Logs attribute group, the event class that is sent once the situation is triggered is ITM_Monitored_Logs.

Note: There are cases where these mappings generate events that are too large for the Tivoli Enterprise Console. In these cases, the event class names and the event slot names are the same, but some of the event slots are omitted.

Each of the event classes is a child of KUL_Base. The KUL_Base event class can be used for generic rules processing for any event from the Monitoring Agent for UNIX Logs.

Table 25. Overview of attribute groups to event classes and slots

Attribute groups	event class and slots
Monitored_Logs	ITM_Monitored_Logs event class with these slots: • managed_system: STRING • log_path: STRING • log_name: STRING • log_type: STRING • log_type_enum: STRING • monitor_status: STRING • monitor_start_per_stop_time: STRING • number_of_events: INTEGER • number_of_events_enum: STRING • number_of_format_errors: INTEGER • number_of_format_errors_enum: STRING • log_size: INTEGER • log_size_enum: STRING • date_last_modified: STRING • debug_mode: STRING • debug_mode_enum: STRING • format_command: STRING • timestamp: STRING • log_path_u: STRING • log_name_u: STRING • log_size_64: INTEGER • log_size_64_enum: STRING • number_of_format_errors_64: INTEGER • number_of_format_errors_64_enum: STRING • number_of_events_64: INTEGER • number_of_events_64_enum: STRING

Table 25. Overview of attribute groups to event classes and slots (continued)

Attribute groups	event class and slots
Log_Entries	ITM_Log_Entries event class with these slots: • managed_system: STRING • log_path: STRING • log_name: STRING • entry_time: STRING • system: STRING • kul_source: STRING • type: STRING • kul_class: STRING • kul_class_enum: STRING • description: STRING • frequency_threshold: INTEGER • period_threshold: INTEGER • timestamp: STRING • log_path_u: STRING • log_name_u: STRING • source_u: STRING • description_u: STRING

Appendix D. Documentation library

This appendix contains information about the publications related to IBM Tivoli Monitoring and to the commonly shared components of Tivoli Management Services. These publications are listed in the following categories:

- IBM Tivoli Monitoring library
- Related publications

See *IBM Tivoli Monitoring and OMEGAMON XE Products: Documentation Guide*, SC23-8816, for information about accessing and using the publications. You can find the *Documentation Guide* in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/. To open the *Documentation Guide* in the information center, select **Using the publications** in the **Contents** pane.

To find a list of new and changed publications, click **What's new** on the Welcome page of the IBM Tivoli Monitoring and OMEGAMON XE Information Center. To find publications from the previous version of a product, click **Previous versions** under the name of the product in the **Contents** pane.

IBM Tivoli Monitoring library

The following publications provide information about IBM Tivoli Monitoring and about the commonly shared components of Tivoli Management Services:

- *Quick Start Guide*

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, GC32-9407

  Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.
- *Program Directory for IBM Tivoli Management Services on z/OS*, GI11-4105

  Gives instructions for the SMP/E installation of the Tivoli Management Services components on z/OS.
- *Configuring the Tivoli Enterprise Monitoring Server on z/OS*, SC27-2313

  Provides instructions for preparing, configuring, and customizing your monitoring servers on z/OS. This guide complements the *IBM Tivoli OMEGAMON XE and IBM Tivoli Management Services on z/OS Common Planning and Configuration Guide* and the *IBM Tivoli Monitoring Installation and Setup Guide*.
- *Administrator's Guide*, SC32-9408

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.

- *High-Availability Guide for Distributed Systems*, SC23-9768

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- Tivoli Enterprise Portal online help

  Provides context-sensitive reference information about all features and customization options of the Tivoli Enterprise Portal. Also gives instructions for using and administering the Tivoli Enterprise Portal.
- *Tivoli Enterprise Portal User's Guide*, SC32-9409

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.
- *Command Reference*, SC32-6045

  Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.
- *Troubleshooting Guide*, GC32-9458

  Provides information to help you troubleshoot problems with the software.
- *Messages*, SC23-7969

  Lists and explains messages generated by all IBM Tivoli Monitoring components and by z/OS-based Tivoli Management Services components (such as Tivoli Enterprise Monitoring Server on z/OS and TMS:Engine).
- *IBM Tivoli Universal Agent User's Guide*, SC32-9459

  Introduces you to the IBM Tivoli Universal Agent, an agent of IBM Tivoli Monitoring. The IBM Tivoli Universal Agent enables you to use the monitoring and automation capabilities of IBM Tivoli Monitoring to monitor any type of data you collect.
- *IBM Tivoli Universal Agent API and Command Programming Reference Guide*, SC32-9461

  Explains the procedures for implementing the IBM Tivoli Universal Agent APIs and provides descriptions, syntax, and return status codes for the API calls and command-line interface commands.
- *Agent Builder User's Guide*, SC32-1921

  Explains how to use the Agent Builder for creating monitoring agents and their installation packages, and for adding functions to existing agents.
- *Performance Analyzer User's Guide*, SC27-4004

  Explains how to use the Performance Analyzer to understand resource consumption trends, identify problems, resolve problems more quickly, and predict and avoid future problems.

Documentation for the base agents

If you purchased IBM Tivoli Monitoring as a product, you received a set of *base* monitoring agents as part of the product. If you purchased a monitoring agent product (for example, an OMEGAMON XE product) that includes the commonly shared components of Tivoli Management Services, you did not receive the base agents.

The following publications provide information about using the base agents.

- Operating system agents:
  - *Windows OS Agent User's Guide*, SC32-9445
  - *UNIX OS Agent User's Guide*, SC32-9446
  - *Linux OS Agent User's Guide*, SC32-9447
  - *i5/OS Agent User's Guide*, SC32-9448
  - *UNIX Log Agent User's Guide*, SC32-9471
- Agentless operating system monitors:
  - *Agentless Monitoring for Windows Operating Systems User's Guide*, SC23-9765
  - *Agentless Monitoring for AIX Operating Systems User's Guide*, SC23-9761
  - *Agentless Monitoring for HP-UX Operating Systems User's Guide*, SC23-9763
  - *Agentless Monitoring for Solaris Operating Systems User's Guide*, SC23-9764
  - *Agentless Monitoring for Linux Operating Systems User's Guide*, SC23-9762
- Warehouse agents:
  - *Warehouse Summarization and Pruning Agent User's Guide*, SC23-9767
  - *Warehouse Proxy Agent User's Guide*, SC23-9766
- System P agents:
  - *AIX Premium Agent User's Guide*, SA23-2237
  - *CEC Base Agent User's Guide*, SC23-5239
  - *HMC Base Agent User's Guide*, SA23-2239
  - *VIOS Premium Agent User's Guide*, SA23-2238
- Other base agents:
  - *Systems Director base Agent User's Guide*, SC27-2872
  - *Tivoli Log File Agent User's Guide*, SC14-7484
  - *Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint User's Guide*, SC32-9490

Related publications

You can find useful information about related products in the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/.

Other sources of documentation

You can also obtain technical documentation about IBM Tivoli Monitoring and related products from the following sources:

- IBM Integrated Service Management Library

  http://www-01.ibm.com/software/brandcatalog/ismlibrary/

  IBM Integrated Service Management Library is an online catalog that contains integration documentation and other downloadable product extensions.
- Redbooks

  http://www.redbooks.ibm.com/

  IBM Redbooks® and Redpapers include information about products from platform and solution perspectives.

- Technotes

  Technotes provide the latest information about known product limitations and workarounds. You can find Technotes through the IBM Software Support Web site at http://www.ibm.com/software/support.
- Tivoli wikis on the IBM developerWorks Web site

  Tivoli Wiki Central at http://www.ibm.com/developerworks/wikis/display/tivoli/Home is the home for interactive wikis that offer best practices and scenarios for using Tivoli products. The wikis contain white papers contributed by IBM employees, and content created by customers and business partners.

  Two of these wikis are of particular relevance to IBM Tivoli Monitoring:

  – Tivoli Distributed Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivolimonitoring/Home provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.
  – Tivoli System z Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivoliomegamon/Home provides information about the OMEGAMON XE products, NetView for z/OS, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.

Appendix E. Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in this product enable users to do the following:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

In addition, the product documentation was modified to include the following features to aid accessibility:

- All documentation is available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Navigating the interface using the keyboard

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Magnifying what is displayed on the screen

You can enlarge information on the product windows using facilities provided by the operating systems on which the product is run. For example, in a Microsoft Windows environment, you can lower the resolution of the screen to enlarge the font sizes of the text on the screen. Refer to the documentation provided by your operating system for more information.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Printed in USA

SC32-9471-05