Tivoli Log File Agent Version 6.3

User's Guide



Tivoli Log File Agent Version 6.3

User's Guide



ote ore using this	information and th	e product it supp	ports, read the in	nformation in "I	Notices" on page	143.

© Copyright IBM Corporation 2010, 2013. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures v	Chapter 6. Workspaces reference 51
	Predefined workspaces
Tables vii	Workspace descriptions
	Log Files Navigator item
Chapter 1 Overview of the egent	LFAProfiles Navigator item 55
Chapter 1. Overview of the agent 1	LogfileEvents Navigator item 55
IBM Tivoli Monitoring	Tivoli Log File Profile subnode
New in this release	iivon 208 ine iioine babiloae i i i i i i i i i i o
Components of the IBM Tivoli Monitoring	Chapter 7 Attributes reference 57
environment 2	Chapter 7. Attributes reference 57
Agent Management Services	Attribute groups for the monitoring agent 57
User interface options	Attributes in each attribute group
Data sources	LFAProfiles attribute group
	Log File RegEx Statistics attribute group 60
Chapter 2. Requirements and agent	Log File Status attribute group 63
installation and configuration 5	LogfileEvents attribute group 67
	LogfileProfileEvents attribute group 71
Requirements for the monitoring agent 5	Performance Object Status attribute group 76
Generate a software product compatibility report	pro Performance Object Status attribute group 81
for the Log File agent 6	Thread Pool Status attribute group 86
Agent-specific installation and configuration 6	Disk capacity planning for historical data 89
Configuring from the configuration panel 6	1 7 1 0
Configuring from the command line 8	Chapter 8. Situations reference 91
Configuration Files 9	Predefined situations
Changing the agent configuration and format files 9	Situation descriptions
Centralized Configuration	
CPU throttling	KLO_Log_Agent_Config_Error situation 92
Remote log file monitoring	KLO_PRO_Log_Agent_Config_Error situation 93
Remote installation and configuration 14	
Starting the agent	Chapter 9. Take Action commands
Forwarding events to Tivoli Netcool OMNIbus	reference
ObjectServers	Predefined Take Action commands 95
Subnodes	
	Chapter 10. Policies reference 97
Chapter 3. Configuration file 23	Predefined policies
onaptor or configuration mo 1 1 1 1 1 20	Tredefined policies
Chapter 4 Formet file 22	Chapter 11 Troubleshooting 00
Chapter 4. Format file	Chapter 11. Troubleshooting 99
Format file specifications	Trace logging
Inheritance	Overview of log file management 100
Multi-line	Principal trace log files
Mappings	Examples: Using trace logs
Value specifiers	RAS trace parameters
Keywords 40	Dynamic modification of trace settings 107
Maximum message length 41	Setting trace parameters for the Tivoli Enterprise
	Console server
Chapter 5. Event filtering and	Problems and workarounds
summarization 43	Installation and configuration troubleshooting 111
	Remote deployment troubleshooting 115
Detecting and filtering duplicate events 43	Agent troubleshooting
Summary interval	CPU throttling troubleshooting
Filtering events	Remote log file monitor troubleshooting 124
Summarization attributes	Workspace troubleshooting
Viewing event filtering and summarization in the	Situation troubleshooting
Tivoli Enterprise Portal	Take Action commands troubleshooting 130
	Support information

Appendix A. Event mapping 131		Other sources of documentation		
Appendix B. Windows 2008 event log	137	Notices		
Appendix C. Documentation library Prerequisite publications.	. 139	Index		

Figures

1.	Historical view and cache view when send_all	3.	Historical view and cache view when
	is selected		send_first is selected 48
2.	Historical view and cache view when	4.	Historical view and cache view when nInteger
	send_none is selected 47		is selected

Tables

1.	Allocation of configuration files to servers	20
2.	Slots and the DEFAULT value	40
3.	EventFloodThreshold values	44
4.	Capacity planning for historical data logged by	
	the Log File agent	90
5.	Information to gather before contacting IBM	
	Software Support	99
6.		101
7.	Problems and solutions for installation and	
	configuration	111
8.	General problems and solutions for	
	uninstallation	113

9.	Remote deployment problems and solutions	115
10.	Agent problems and solutions	116
11.	CPU throttling problems and solutions	123
12.	Remote log file monitor problems and	
	solutions	124
13.	Workspace problems and solutions	125
14.	Situation problems and solutions	127
15.	Take Action commands problems and	
	solutions	130

Chapter 1. Overview of the agent

The Tivoli Log File Agent (product code LO) provides you with the capability to monitor Application log files.

IBM® Tivoli® Monitoring is the base software for the Log File agent.

The Tivoli Log File Agent is an agent that provides a configurable log file monitoring capability that uses regular expressions. For compatibility, the agent can consume the configuration information and format strings previously used by the Tivoli Event Console Log File Adapter. These strings allow the agent to filter the log data according to patterns in the format file, and submit only the interesting data to an event consumer. The agent can send data both to a Tivoli Enterprise Monitoring Server or through the Event Integration Facility (EIF) to any EIF receiver, such as the OMNIbus EIF probe.

IBM Tivoli Monitoring

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. It also provides useful historical data that you can use to track trends and to troubleshoot system problems.

You can use IBM Tivoli Monitoring to achieve the following tasks:

- 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 take actions, schedule work, and automate manual tasks.

The Tivoli Enterprise Portal is the interface for IBM Tivoli Monitoring products. You can use the consolidated view of your environment as seen in the Tivoli Enterprise Portal to monitor and resolve performance issues throughout the enterprise.

See the IBM Tivoli Monitoring publications listed in "Prerequisite publications" on page 139 for complete information about IBM Tivoli Monitoring and the Tivoli Enterprise Portal.

New in this release

For version 6.3 of the Log File agent, the following enhancements were made since version 6.2.3.2:

- Remote log file monitoring. The Log File agent can monitor text logs on remote systems by using the Secure Shell (SSH) File Transfer Protocol. For more information, see "Remote log file monitoring" on page 11.
- Changes related to system requirements. See the information about system requirements in Software product compatibility reports (http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html). For more information about running a software product compatibility report specific to Log File agent, see "Generate a software product compatibility report for the Log File agent" on page 6.
- Updated klo.baroc file to support IBM Tivoli Enterprise Console[®] event mapping changes.
- Added support for the IBM Prerequisite Scanner. The Prerequisite Scanner is a stand-alone prerequisite
 checking tool that analyzes system environments before the installation or upgrade of a Tivoli product
 or IBM solution.

Components of the IBM Tivoli Monitoring environment

After you install and set up the Log File agent, you have an environment that contains the client, server, and monitoring agent implementation for Tivoli Monitoring.

This Tivoli Monitoring environment contains the following components:

Tivoli Enterprise Portal client

The portal has a user interface based on Java[™] for viewing and monitoring your enterprise.

Tivoli Enterprise Portal Server

The portal server is placed between the client and the Tivoli Enterprise Monitoring Server and enables retrieval, manipulation, and analysis of data from the monitoring agents. The Tivoli Enterprise Portal Server is the central repository for all user data.

Tivoli Enterprise Monitoring Server

The monitoring server acts as a collection and control point for alerts received from the monitoring agents, and collects their performance and availability data. The Tivoli Enterprise Monitoring Server is also a repository for historical data.

Tivoli Enterprise Monitoring Agent, Log File agent

This monitoring agent collects data and distributes the data to the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, Tivoli Enterprise Portal, Tivoli Data Warehouse, and IBM Dashboard Application Services Hub.

Multiple copies of this agent can run on the same system.

IBM Tivoli Netcool/OMNIbus

Tivoli Netcool/OMNIbus is an optional component and the recommended event management component. The Netcool/OMNIbus software is a service level management (SLM) system that delivers real-time, centralized monitoring of complex networks and IT domain events. Event information is tracked in a high-performance, in-memory database and presented to specific users through individually configurable filters and views. The software includes automation functions that you can use to perform intelligent processing on managed events. You can use this software to forward events for Tivoli Monitoring situations to Tivoli Netcool/OMNIbus.

IBM Tivoli Enterprise Console

The Tivoli Enterprise Console is an optional component that acts as a central collection point for events from various sources, including events from other Tivoli software applications, Tivoli partner applications, custom applications, network management platforms, and relational database systems. You can view these events through the Tivoli Enterprise Portal (by using the event viewer), and you can forward events from Tivoli Monitoring situations to the Tivoli Enterprise Console component. If you do not already use Tivoli Enterprise Console and need an event management component, you can choose to use IBM Tivoli Netcool/OMNIbus.

IBM Tivoli Common Reporting

Tivoli Common Reporting is a separately installable feature available to users of Tivoli software that provides a consistent approach to generating and customizing reports. Some individual products provide reports that are designed for use with Tivoli Common Reporting and have a consistent look and feel.

IBM Tivoli Application Dependency Discovery Manager (TADDM)

TADDM delivers automated discovery and configuration tracking capabilities to build application maps that provide real-time visibility into application complexity.

IBM Tivoli Business Service Manager

The Tivoli Business Service Manager component delivers real-time information to help you respond to alerts effectively based on business requirements. Optionally, you can use this component to meet service-level agreements (SLAs). Use the Tivoli Business Service Manager tools to help build a service model that you can integrate with Tivoli Netcool/OMNIbus alerts or

optionally integrate with data from an SQL data source. Optional components provide access to data from other IBM Tivoli applications such as Tivoli Monitoring and TADDM.

IBM Dashboard Application Services Hub

The Dashboard Application Services Hub has a core set of components that provide such administrative essentials as network security and database management. This component replaces the Tivoli Integrated Portal component after version 2.2.

Agent Management Services

You can use IBM Tivoli Monitoring Agent Management Services to manage the Log File agent.

Agent Management Services is available for the following IBM Tivoli Monitoring OS agents: Windows, Linux, and UNIX. The services are designed to keep the Log File agent available, and to provide information about the status of the product to the Tivoli Enterprise Portal. IBM Tivoli Monitoring V6.2.2, Fix Pack 2 or later provides support for Agent Management Services. For more information about Agent Management Services, see Agent Management Services in the IBM Tivoli Monitoring Administrator's Guide.

User interface options

Installation of the base IBM Tivoli Monitoring software and other integrated applications provides various interfaces that you can use to work with your resources and data.

The following interfaces are available:

Tivoli Enterprise Portal user interface

You can run the Tivoli Enterprise Portal as a desktop application or a browser application. The client interface is a graphical user interface (GUI) based on Java on a Windows or Linux workstation. The browser application is automatically installed with the Tivoli Enterprise Portal Server. The desktop application is installed by using the Tivoli Monitoring installation media or with a Java Web Start application. To start the Tivoli Enterprise Portal browser client in your Internet browser, enter the URL for a specific Tivoli Enterprise Portal browser client installed on your Web server.

Command-line interface

You can use Tivoli Monitoring commands to manage the Tivoli Monitoring components and their configuration. You can also run commands at the Tivoli Enterprise Console event server or the Tivoli Netcool/OMNIbus ObjectServer to configure event synchronization for enterprise situations.

Manage Tivoli Enterprise Monitoring Services window

You can use the window for the Manage Tivoli Enterprise Monitoring Services utility to configure the agent and start Tivoli services not designated to start automatically.

IBM Tivoli Netcool/OMNIbus event list

You can use the Netcool/OMNIbus event list to monitor and manage events. An event is created when the Netcool/OMNIbus ObjectServer receives an event, alert, message, or data item. Each event is made up of columns (or fields) of information that are displayed in a row in the ObjectServer alerts.status table. The Tivoli Netcool/OMNIbus web GUI is also a web-based application that processes network events from one or more data sources and presents the event data in various graphical formats.

IBM Tivoli Enterprise Console

You can use the Tivoli Enterprise Console to help ensure the optimal availability of an IT service for an organization. The Tivoli Enterprise Console is an event management application that integrates system, network, database, and application management. If you do not already use Tivoli Enterprise Console and need an event management component, you can choose to use Tivoli Netcool/OMNIbus.

IBM Tivoli Common Reporting

Use the Tivoli Common Reporting web user interface for specifying report parameters and other report properties, generating formatted reports, scheduling reports, and viewing reports. This user interface is based on the Dashboard Application Services Hub.

IBM Tivoli Application Dependency Discovery Manager

The Discovery Management Console is the TADDM client user interface for managing discoveries.

IBM Tivoli Business Service Manager

The Tivoli Business Service Manager console provides a graphical user interface that you can use to logically link services and business requirements within the service model. The service model provides an operator with a second-by-second view of how an enterprise is performing at any moment in time or how the enterprise performed over a time period.

IBM Dashboard Application Services Hub

The Dashboard Application Services Hub provides an administrative console for applications that use this framework. It is a web-based console that provides common task navigation for products, aggregation of data from multiple products into a single view, and the passing of messages between views from different products. This interface replaces the Tivoli Integrated Portal component after version 2.2.

Data sources

Monitoring agents collect data from specific data sources.

The Log File agent collects data from the following sources:

Log files

The agent uses the file system to monitor application log files or other data files to gather metrics.

Chapter 2. Requirements and agent installation and configuration

Agent installation and configuration requires the use of the *IBM Tivoli Monitoring Installation and Setup Guide* and agent-specific installation and configuration information.

To install and configure Tivoli Log File Agent, use the procedures for installing monitoring agents in the *IBM Tivoli Monitoring Installation and Setup Guide* along with the agent-specific installation and configuration information.

If you are installing silently by using a response file, see *Performing a silent installation of IBM Tivoli Monitoring in the IBM Tivoli Monitoring Installation and Setup Guide*.

With the self-describing agent capability, new or updated IBM Tivoli Monitoring agents using IBM Tivoli Monitoring V6.2.3 or later can become operational after installation without having to perform additional product support installation steps. To take advantage of this capability, see *Enabling self-describing agent capability at the hub monitoring server* in the *IBM Tivoli Monitoring Installation and Setup Guide*. Also, see *Self-describing monitoring agents* in the *IBM Tivoli Monitoring Administrator's Guide*.

Requirements for the monitoring agent

The Log File agent has specific operating system, resource, and software requirements.

In addition to the requirements described in the *IBM Tivoli Monitoring Installation and Setup Guide*, the Tivoli Log File Agent has the following requirements:

- For the most current information about system requirements, see the Software product compatibility reports (http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html). For more information about running a software product compatibility report specific to Log File agent, see "Generate a software product compatibility report for the Log File agent" on page 6.
- A single computer that hosts the hub monitoring server, portal server, and a monitoring agent requires
 approximately 300 MB of space. A computer that hosts only the monitoring agent requires
 approximately 30 MB of space, including the specific enablement code for the monitoring agent. More
 space is required for each additional monitoring agent that you deploy on the monitoring computer.
 The 30 MB figure is approximate and varies by operating system.
- Linux versions require some compatibility libraries to be installed for the agent to work correctly. The latest versions of libstdc++, libgcc, and compat-libstdc++ are required for the agent to run correctly. Linux RedHat 4, 5 and 6, and SuSE 9 and 10 also require the C++ Runtime 6.0, libstdc++.so.6.
- The monitoring agent must be connected to one (or both) of the following software products:
 - IBM Tivoli Monitoring V6.2.2.2 or later
 - Tivoli Netcool/OMNIbus

After you install the prerequisite software, install the following software, that is required for the Tivoli Log File Agent to operate:

- Tivoli Log File Agent

To send events to IBM Tivoli Monitoring, install the following software:

- Tivoli Log File Agent for Tivoli Enterprise Monitoring Server support
- Tivoli Log File Agent for Tivoli Enterprise Portal Server support
- Tivoli Log File Agent for Tivoli Enterprise Portal Desktop Client support
- Tivoli Log File Agent for Tivoli Enterprise Portal Browser Client support

To send events to Tivoli Netcool/OMNIbus, install the following software:

Probe for Tivoli EIF

For more information, see the IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF Reference Guide.

This monitoring agent collects data from the following data sources:

Log files

The agent uses the file system to monitor the following log file types:

- · Application log files
- · Windows event logs
- UNIX system logs
- AIX® error report command log

Generate a software product compatibility report for the Log File agent

Use this procedure to generate a Software product compatibility report for the Log File agent.

About this task

Software Product Compatibility Reports are available on the IBM Support Portal. The Software Product Compatibility Reports is a tool designed to allow you to easily generate custom reports about compatible IBM software combinations

Procedure

- 1. Go to the Software Product Compatibility Reports site at Software product compatibility reports.
- 2. Select Create a report under In-depth reports.
- 3. Enter Tivoli Monitoring in the Full or partial product name: field.
- 4. Click the search icon.
- 5. Select **Tivoli Monitoring** from the **Search results:** area.
- 6. Select a version from the Version list.
- 7. Under Scope of report, select Product component.
- 8. Under Select component, select Log File Agent.
- 9. Click **Submit**. The report is displayed in a new browser window.

Agent-specific installation and configuration

In addition to the installation and configuration information in the *IBM Tivoli Monitoring Installation and Setup Guide*, use the information in this section to install and configure the Tivoli Log File Agent.

You configure your agent by using the configuration panel or command line and by creating configuration files. The configuration files tell the agent what log files to monitor, where to find them and what formatting to use.

Configuring from the configuration panel

Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Procedure

Using a Tivoli Enterprise Monitoring Server

• To configure the agent from the Manage Tivoli Enterprise Monitoring Services graphical interface, select the agent from the list, right-click, and choose **Configure** on UNIX/Linux, or **Configure Using Defaults** on Windows. The configuration panel opens. The configuration panel has two tabs, one is named **Log File Adapter Configuration** and the other is named **Log File Adapter Global Settings**.

• In the Log File Adapter Configuration tab, the following configuration items must be specified:

- Instance Name

Enter an identifier string that is meaningful to you to uniquely identify this instance of the agent. It is possible to run more than one instance of the agent on a particular system, each with a separate configuration. For example, you might want to monitor logs from two different applications on one system. Because each application formats its log differently, you must provide separate format files for each of them. Provide a name here that identifies which of the two applications this instance of the agent is to monitor.

- Conf file

Full path name to the .conf file that contains the detailed configuration options. For example, if you were previously using the Tivoli Enterprise Console log file adapter, on a UNIX system, your configuration file might be /etc/Tivoli/tecad/etc/tecad_logfile.conf, and that is the value you provide here. For more information about the .conf file, see Chapter 3, "Configuration file," on page 23.

Note: If you use the subnode feature to create monitoring profiles, you can have more than one configuration file. If you have more than one configuration file, do not enter a path name here. For more information, see "Subnodes" on page 18.

- Format file

Full path name to the .fmt file that contains the format string patterns that you want to match in the monitored log. For example, if you were previously using the Tivoli Enterprise Console log file adapter, on a UNIX system, your format file might be /etc/Tivoli/tecad/etc/C/tecad_logfile.fmt, and that is the value you would provide here. For more information about the .fmt file, see Chapter 4, "Format file," on page 33.

Note: If you use the subnode feature to create monitoring profiles, you can have more than one format file. If you have more than one format file, do not enter a path name here. For more information, see "Subnodes" on page 18.

- Send ITM Events

If this option is set to Yes the agent sends event data to a Tivoli Enterprise Monitoring Server, and if the option is set to No the agent does not send the event data. The setting of this option is global and applies to all monitoring profiles.

- Send EIF Events

If this option is set to Yes the agent sends event data directly to a receiver through EIF, and if set to No it does not send the data. The setting of this option is global and applies to all monitoring profiles.

Note: The EIF receiver must consume events otherwise there can be problems when the agent cache fills. For more information about buffering of EIF events in the agent cache, see "The following options apply only where events are being forwarded to EIF: " on page 30.

- Automatically initialize UNIX syslog

If this option is set to Yes on UNIX and Linux systems, the agent automatically configures the syslog facility to write a standard set of events to a pipe that the agent monitors. By enabling this setting, you can monitor syslog events without maintaining and rolling-over log files. If set to USE_CONF_FILE_VALUE, the agent checks the configuration (.conf) file for the setting (AutoInitSyslog). If not set in the configuration file, it is the same as being set to No.

This setting is ignored on Microsoft Windows. The default value is USE CONF FILE VALUE.

- In the **Log File Adapter Global Settings** tab, the following optional configuration items can be specified:
 - Process Priority Class

This setting is the operating system scheduler priority for the process. A is lowest, C is the operating system default, and F is the highest priority. The setting is one of the following values: A, B, C, D, E, F, "USE_CONF_FILE_VALUE". The default setting is USE_CONF_FILE_VALUE. This setting is associated with the CPU throttling feature.

- Process maximum CPU percentage

This setting is the maximum allowable percentage of all system CPU that the agent can use over a 1-minute interval. Valid values are from 5 - 100. The default value is 100. This setting is associated with the CPU throttling feature. If you specify a value less than 5, the minimum value of 5 is used.

- Configuration file autodiscovery directory

This setting is a string value that sets the point of the file system directory tree to scan for autodiscovery of configuration (.conf) files. There must be a format (.fmt) file with the same base name in the same location. A monitoring profile is created for each such pair of files found. The default setting is \${CANDLE_HOME}/config/lo. This setting is associated with the subnodes (monitoring profiles) feature. Only one directory path can be specified, you cannot for example enter multiple directory paths separated by commas.

Note: To avoid impacting system resources, do not set this path at too high a level in your directory tree. The agent searches recursively from the point specified. For example, if you set the path to / the agent attempts to search your entire file system every minute.

Not using a Tivoli Enterprise Monitoring Server

• On the Configuration panel described previously, set the **Send ITM Events** option to No.

Note: On non-Windows systems, the Manage Tivoli Enterprise Monitoring Services interface also provides a panel for configuring the connection to a Tivoli Enterprise Monitoring Server. The panel has a check box labeled **No TEMS**. Select this option, and the **TEMS Hostname** field is disabled.

Related concepts:

"CPU throttling" on page 10

Use CPU throttling to manage the processor usage of an agent.

"Subnodes" on page 18

Use subnodes to implement monitoring profiles that monitor multiple applications from a single agent instance.

Related tasks:

"Forwarding events to Tivoli Netcool OMNIbus ObjectServers" on page 15

You can configure the Tivoli Log File Agent to forward events to one or more Tivoli Netcool/OMNIbus ObjectServers.

"Configuring from the command line"

You can configure the Tivoli Log File Agent from the command line on UNIX systems

Related reference:

"Configuration Files" on page 9

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

Configuring from the command line

You can configure the Tivoli Log File Agent from the command line on UNIX systems

Procedure

To configure the Tivoli Log File Agent from the command line on UNIX systems, type the following command and follow the resulting prompts:

itmcmd config -A lo

Note: The configuration options are the same as the options described for configuring from the configuration panel.

Related tasks:

"Configuring from the configuration panel" on page 6

Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Configuration Files

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

About the Configuration Files

Agent configuration indicates the naming, the mode of agent operation, and the location of the configuration (.conf) file and the format (.fmt) file. The two configuration files control the file monitoring, and their location must be specified during the IBM Tivoli Monitoring configuration process. Subsequent configuration is drawn from these files.

Note: If you use the subnode feature to create monitoring profiles, you can have more than one pair of configuration (.conf) and format (.fmt) files. For more information, see "Subnodes" on page 18.

Example configuration and format files are provided with the agent, for example regex1.conf and regex1.fmt are configuration and format files for use with regular expressions. The example files can be found in the examples directory on the agent media image.

The path and file names of the (.conf) and (.fmt) files must be ASCII only. A useful convention is to name your configuration and format files to match the application or log file you want to monitor.

Note: Configuration file Filter statements apply only to EIF. The Filter statements that can be used in the .conf file, apply to any EIF events that are sent to an EIF receiver, but do not apply to events sent to a Tivoli Enterprise Monitoring Server. Therefore, if filtering of events to the Tivoli Enterprise Monitoring Server is important, use the format file. For example, assigning events in the format file to the special "*DISCARD*" class causes them to be ignored. This mechanism is honored through either event channel.

Related concepts:

Chapter 4, "Format file," on page 33

The Tivoli Log File Agent extracts information from system log messages and then matches different log messages to event classes. A format file serves as a lookup file for matching log messages to event classes, telling the event class what to read, what to match, and how to format the data.

"Format file specifications" on page 33

The format file describes the patterns that the agent looks for to match events in the monitored logs. The format file consists one or more format specifications.

Related tasks:

"Configuring from the configuration panel" on page 6

Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Related reference:

Chapter 3, "Configuration file," on page 23

The Tivoli Log File Agent uses a configuration file that contains configuration options and filters. You create this configuration file and configure the agent instance to use it.

Changing the agent configuration and format files

The Tivoli Log File Agent reads its configuration (.conf) and format (.fmt) files when it starts, and monitors their timestamp every 60 seconds thereafter.

If the timestamp of the configuration or format file changes, the agent reinitializes its configuration dynamically, without requiring a restart. During reinitialization monitoring is interrupted momentarily. When monitoring resumes, the agent must determine the position in the monitored logs from which to restart. In this situation, the agent behaves in the same way as a full stop and restart.

Note: Agent reinitialization after a configuration or format file change, resets information in the Log File RegEx Statistics, Log File Status, and Log File Event attribute groups. For more information about these attribute groups, see Chapter 7, "Attributes reference," on page 57.

By default, the agent starts monitoring from the end of the file, when the reinitialization completes. This starting position can cause events that occurred during the interruption of monitoring to be missed. To ensure that such events are picked up when monitoring resumes, use the NumEventsToCatchUp=-1 setting.

Setting NumEventsToCatchUp=-1 causes a position file to be maintained. The position file is updated each time the agent reads the log file. The update saves the position of the agent in the log file, in case of an agent restart. Maintaining the position file has a small performance impact, and so maintain this file only if required. For more information about the NumEventsToCatchUp, see Chapter 3, "Configuration file," on page 23.

Note: Some configuration values are not present in the configuration file and are set during initial configuration. If you change these values, you must restart the agent.

Dynamic reinitialization of the agent configuration enables IBM Tivoli Monitoring Centralized Configuration, for management of configuration and format files. For more information, see "Centralized Configuration"

Related concepts:

"Format file specifications" on page 33

The format file describes the patterns that the agent looks for to match events in the monitored logs. The format file consists one or more format specifications.

Related reference:

Chapter 3, "Configuration file," on page 23

The Tivoli Log File Agent uses a configuration file that contains configuration options and filters. You create this configuration file and configure the agent instance to use it.

Centralized Configuration

Use Centralized Configuration to update local configuration files on many monitoring agents.

You can use Centralized Configuration to host the Log File agent configuration and format files on a central http server. A user-provided XML file installed with the Tivoli Log File Agent describes where the http server is located, the paths to the files, and authentication information (if any). The agent then polls the server periodically, and downloads the files any time there are changes. The agent automatically notices that the local copies of the files changed, and reinitializes itself to use them.

For full setup details, see Centralized Configuration in the IBM Tivoli Monitoring Administrator's Guide.

CPU throttling

Use CPU throttling to manage the processor usage of an agent.

When installing the agent, the optional Process Priority Class and Process maximum CPU percentage settings can be specified. You can use these optional settings to limit the use of processor time by the agent. Using these settings you can: set the priority, or limit the processor usage, of the Log File agent process. Generally you set one or the other.

Use the **Process Priority Class** setting to set the operating system priority of the Log File agent process. Set this value correctly, and the Log File agent runs only when other processes on the system do not need the processor. Also when the value is set correctly the Log File agent processes logs only when it does not interfere with other running applications. Use the **Process Priority Class** value when you do not want to artificially limit the Log File agent to a certain percentage of processor time. For example, when the system is idle.

Use the **Process maximum CPU percentage** setting to manage the maximum processor usage of the Log File agent process over a 1-minute interval. If a **Process maximum CPU percentage** value less than 100 is specified, throttling is in effect. The **Process maximum CPU percentage** value sets a maximum limit for the agent's use of the processor over a 1-minute time interval.

When setting the **Process maximum CPU percentage** value, you must be aware of the following points:

- The **Process maximum CPU percentage** value is a goal over a 1-minute measurement interval. For intervals less than 1 minute the agent can use 100% of the processor time and still meet its goal for the 1-minute measurement interval. By way of comparison, a Linux installation can use the top command to show the processor usage of a process and show values that change every second. But the top command with the interval of 60, top -d 60, is a better way to assess the processor usage of the process over a 1-minute interval.
- The **Process maximum CPU percentage** value is a goal over a 1-minute measurement interval. Actual usage can be less. For example, if in the second half of a 1-minute interval, many events arrive, the agent does not consider that the first half of the interval was under-used and therefore does not exceed the maximum limit in the second half of the interval.
- CPU throttling might not behave as expected when the agent is run in a virtual machine (VM). The clock of a VM is accurate, but the amount of actual processor time allocated to the VM cannot be determined from inside the VM. That is, the amount of actual processor time available in a specific amount of real time can vary. This variation can cause the agent's view of its processor usage to be inaccurate and can cause the agent to miss its goals.
- CPU throttling extends the elapsed time it takes to process large amounts of data. A certain amount of processor time is required to process events. If the limit is set too low, the agent might not keep up with high event rates.

Related tasks:

"Configuring from the configuration panel" on page 6 Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Remote log file monitoring

You can use Remote log file monitoring to monitor log files on remote systems.

Log File agent can monitor text log files on remote systems by using the Secure Shell (SSH) File Transfer Protocol. SSH File Transfer Protocol is also known as Secure FTP, or SFTP. Remote log file monitoring requires that the remote systems have a working SSH daemon that supports the SFTP protocol. For each Log File agent monitoring profile, you can specify a list of remote hosts to monitor. You specify the list of remote hosts by using a configuration (.conf) file setting, *SshHostList* in the configuration file of the monitoring profile. For more information about monitoring profiles, see "Subnodes" on page 18. The log files to be monitored are configured by using either the *LogSources* or *RegexLogSources* configuration settings as they are normally used for non-remote files. When you specify a list of remote hosts by using the *SshHostList* setting, all configured log files are monitored on each of the listed remote hosts.

Authentication information for the remote hosts must also be provided. The information consists of a user name, which is valid on all the remote systems, and a password or a private key-public key pair for certificate-based authentication. All remote systems within a single configuration must use the same authentication data.

It is important to note that regular expression processing can be processor-intensive. If a single agent is being used to monitor log files on multiple remote systems, that agent runs all regular-expression processing. It runs this processing on the data that arrives from all of the remote hosts. This process centralizes the workload rather than distributing it. This centralization means that as the number of remote systems that are being monitored increases, the workload on the system on which the agent runs can become substantial. In most cases, it is wise to dedicate a server to the task of running the monitoring agent. This architecture also has the advantage that the systems that are being monitored are spared the processor load of regular expression processing. In this way, regular expression processing does not interfere with the business purpose of those systems.

When an event is generated and sent to Tivoli Monitoring, it contains an attribute, **RemoteHost**, which is an identifier of the system on which the event originated. If the event originated on the local system, this attribute is empty. If the event originated on a remote system, it contains a string of the form user@host:port, which indicates the remote host name on which the event occurred, and the user and port on that host that are used to connect.

When an event is generated and sent to Tivoli Netcool/OMNIbus by using the Event Integration Facility (EIF), a slot that is named **RemoteHost** is used to convey information about the remote host. Before you send events to Tivoli Netcool/OMNIbus, you must define this slot in the format (.fmt) file. You name the slot **RemoteHost**, and give it the value *DEFAULT*. The Log File agent fills the value with information about the remote origin of the event. If the event originated on the local system, this attribute is empty. If the event originated on a remote system, it contains a string of the form user@host:port, which indicates the remote host name on which the event occurred, and the user and port on that host that are used to connect. For more information about the DEFAULT keyword, see "Keywords" on page 40.

The Monitored File Status attribute group for Tivoli Self-Monitoring now contains a similar attribute. When you use this attribute, for each monitored file you see which remote system the file is being monitored on. The other information in this attribute group is not affected when the file is remote. For more information about configuration settings for remote log file monitoring, see "Options for remote log file monitoring by using SSH" on page 27.

Example

Monitor three remote hosts, called host1, host2, and host3, and send events by using the Event Integration Facility (EIF), including the remote origin of the event in the event. Authenticate on the remote host as a user called *loguser*, by using that user's password.

Add the following to the configuration (.conf) file (in addition to other options as usual):

SshHostList=host1,host2,host3 SshUserid=loguser SshAuthType=PASSWORD SshPassword=password

In the format file, add the required remote slot to our base event, and allow other events to inherit it:

```
REGEX BaseRemoteEvent
^(.*)$
RemoteHost DEFAULT
msg $1
END
REGEX ErrorRemoteEvent FOLLOWS BaseRemoteEvent
^ERROR:(.*)$
END
```

Supported SSH capabilities and features

The following are the SSH capabilities and features that are supported by the Log File agent. The remote system must support one of the Key Exchange Methods, Hostkey Types, Ciphers, Compression, and MAC hashes for the local / remote system communications. For example, if the remote system SSH daemon does not support one of the Key Exchange Methods, the Log File Status object shows a status of NO SUPPORTED DESC. :

Key Exchange Methods

diffie-hellman-group1-sha1, diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1

Hostkey Types

ssh-rsa, ssh-dss

Ciphers

aes256-cbc (rijndael-cbc@lysator.liu.se), aes192-cbc, aes128-cbc, arcfour, 3des-cbc

Compression

None

MAC hashes

hmac-sha1, hmac-sha1-96, hmac-md5, hmac-md5-96

Note: As Windows does not provide SSH support natively, if you want to use a Windows system as the remote server, you must install and configure SSH software on it.

Encrypting a password or pass phrase

For increased security, you can encrypt passwords and pass phrases that are transmitted to remote systems when you use Remote log file monitoring.

About this task

The encrypted password and pass phrases are stored in the configuration (.conf) file. For more information about the configuration file, see Chapter 3, "Configuration file," on page 23.

Procedure

Run the itmpwdsnmp command and supply the password or pass phrase that is to be encrypted:

• On UNIX and Linux systems, the command is run from the IBM Tivoli Monitoring installation directory:

Restriction: You must set the CANDLEHOME environment variable to the installation directory path.

On Windows, the command is run from the TMAITM6 directory (C:\IBM\ITM\TMAITM6 by default).

Example of the command when it is run on a Linux system:

```
$ /opt/IBM/ITM/bin/itmpwdsnmp
Enter string to be encrypted:
mypassword
Confirm string:
mypassword
{AES256:keyfile:a}Z7BS23aupYqwlXb1Gh+weg==
```

\$ export CANDLEHOME=/opt/IBM/ITM

The entire output from the command, {AES256:keyfile:a}Z7BS23aupYqwlXb1Gh+weg== in the example, is used to set the SshPassword in the agent configuration file. The prefix {AES256:keyfile:a} denotes to the

agent that the password is encrypted.

To encrypt a pass phrase for a private key file, follow the same procedure.

Remote installation and configuration

You can install the monitoring agent remotely from the Tivoli Enterprise Portal or from the command line.

When installing the agent remotely, you must provide the configuration values for the agent to operate. See "Configuring from the configuration panel" on page 6.

See the **tacmd describeSystemType** section in the *IBM Tivoli Monitoring Command Reference* for information about the configuration options that are available to use with the **configureSystem** or **addSystem** commands.

You must add the agent to the remote deploy depot by using the **tacmd addBundles** command before the **addsystem** command can be issued. For example:

```
tacmd addbundles -i rath to bundles> -t lo -f
```

You must also run the **tacmd addBundles** command in order to remotely configure the agent from the Tivoli Enterprise Portal or from the command-line interface, and to remotely start and stop agent instances.

To install the agent from the Tivoli Enterprise Portal, see the IBM Tivoli Monitoring Installation and Setup Guide.

If you are using the command-line interface, the following are examples of remote installation and configuration commands:

Single configuration example

Windows operating systems with a single configuration. This example is a configuration that sends events to a Tivoli Netcool/OMNIbus server through the Probe for Tivoli EIF.

```
tacmd addSystem -t LO -n Primary:sample.node.name:NT -p LOG_FILE_ADAPTER_CONFIGURATION.KLO_AUTO_INIT_SYSLOG=No LOG_FILE_ADAPTER_CONFIGURATION.KLO_CONF_FILE=value LOG_FILE_ADAPTER_CONFIGURATION.KLO_FORMAT_FILE=value LOG_FILE_ADAPTER_CONFIGURATION.KLO_SEND_EIF_EVENTS=Yes LOG_FILE_ADAPTER_CONFIGURATION.KLO_SEND_ITM_EVENTS=No INSTANCE="inst1"
```

Multiple configuration example

UNIX operating systems that use autodiscovery of multiple configuration files in the directory /opt/IBM/ITM/config/lo to create subnodes. This example is a configuration that sends events to Tivoli Monitoring, and is limited to 25% processor utilization.

```
tacmd addSystem -t LO -n sample.node.name:KUX
-p LOG_FILE_ADAPTER_CONFIGURATION.KLO_AUTO_INIT_SYSLOG=USE_CONF_FILE_VALUE
LOG_FILE_ADAPTER_CONFIGURATION.KLO_SEND_EIF_EVENTS=No
LOG_FILE_ADAPTER_CONFIGURATION.KLO_SEND_ITM_EVENTS=Yes
LOG_FILE_ADAPTER_GLOBAL_SETTINGS.KLO_FILE_DISCOVERY_DIR=/opt/IBM/ITM/config/lo
LOG_FILE_ADAPTER_GLOBAL_SETTINGS.KLO_PROCESS_MAX_CPU_PCT=25
INSTANCE="inst2"
```

Starting the agent

Use this procedure to start the Tivoli Log File Agent from the command line on UNIX.

To start the Log File agent from the command line on UNIX, type the following command: itmcmd agent -o *instance* start lo

Where: *instance* specifies the label that you supplied during configuration.

To start an instance of the Log File agent from the GUI on UNIX, follow these steps:

- 1. Type the bin/CandleManage command from your IBM Tivoli Monitoring installation directory to start the GUI.
- 2. Right-click on the instance.
- 3. Choose **start**. If you configured one or more instances, the GUI shows a panel with a list of the instances. When prompted, select the instance to start.

To start an instance of the Log File agent on Windows, follow these steps:

- 1. From the Start menu, specify Start->IBM Tivoli Monitoring->Manage Tivoli Monitoring Services.
- 2. Right-click on the instance.
- **3**. Choose **start**. If you configured one or more instances, the GUI shows a panel with a list of the instances. When prompted, select the instance to start.

Forwarding events to Tivoli Netcool OMNIbus ObjectServers

You can configure the Tivoli Log File Agent to forward events to one or more Tivoli Netcool/OMNIbus ObjectServers.

Procedure

- 1. Configure the IBM Tivoli Netcool/OMNIbus ObjectServer to receive Tivoli Enterprise Console events.
- 2. Install the Probe for Tivoli EIF as described in the IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF Reference Guide.
- 3. Run the Probe for Tivoli EIF.
- 4. Configure the Tivoli Log File Agent to forward events to the Probe for Tivoli EIF listener port by setting the *Send EIF Events to OMNIbus* value to Yes.
- 5. Optional: To forward events to more than one Netcool® OMNIbus ObjectServer, add an entry to the configuration file to indicate the number of servers: *NumAdditionalServers=n*, where n is a positive integer not greater than 8.
- 6. Configure the location of the primary EIF Probe by setting the *ServerLocation* and *ServerPort* variables in the .conf file. *ServerLocation* can be either a host name or IP address. *ServerPort* is the network port number that your EIF probe is listening on, For example, ServerLocation=probehost.mycompany.com and ServerPort=5529.

If you completed the previous optional step to forward events to more than one server, you can specify the additional servers by using the same configuration names, but with an index number appended, starting at 1, for example:

BufEvtPath=/tmp/main.cache ServerLocation=1.2.3.4 ServerPort=1000 NumAdditionalServers=2 BufEvtPath1=/tmp/alt1.cache ServerLocation1=2.3.4.5 ServerPort1=1000 BufEvtPath2=/tmp/alt2.cache ServerLocation2=3.4.5.6 ServerPort2=3000 In this example, the configuration causes the agent to send EIF events to three receivers simultaneously: the default receiver and two alternates. The default receiver must always be configured before the alternates can be used. The alternates are optional. If you want to, you can specify failover values for the server locations, such as: ServerLocation1=2.3.4.5,2.3.4.6. Only the BufEvtPath, ServerLocation, and ServerPort variables can have different values for the alternative configurations. All other values are copied over directly and are the same on all receivers.

7. Start the Tivoli Log File Agent, see "Starting the agent" on page 15 for more information.

Related tasks:

"Configuring from the configuration panel" on page 6

Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Configuring the ObjectServer

Configure the Tivoli Netcool/OMNIbus ObjectServer to map Log File agent custom fields to Tivoli Netcool/OMNIbus ObjectServer fields.

Before you begin

Bring up the ObjectServer. For example: \$OMNIHOME/bin/nco objserv -name NCOMS

About this task

This section describes how to configure the ObjectServer to map Log File agent custom fields to Tivoli Netcool/OMNIbus ObjectServer fields.

To map Log File agent custom fields to IBM Tivoli Tivoli Netcool/OMNIbus ObjectServer fields, you must add some extra custom fields to the ObjectServer.

Procedure

• On UNIX operating systems, run the following command:

```
$OMNIHOME/bin/nco sql -user username -password password
-server server_name < path_to_file/tec_db_update.sql</pre>
```

for example:

```
$NCHOME/omnibus/bin/nco sql -user root -password "" -server NCOMS
< $NCHOME/omnibus/probes/linux2x86/tec_db_update.sql</pre>
```

• On Windows, run the following command:

```
"%OMNIHOME%"\bin\redist\isql.exe -S server name -U username -P password
< "%OMNIHOME%"\probes\win32\tec db update.sql
```

for example:

```
"%OMNIHOME%"\bin\redist\isql.exe -S NCOMS -U root -P ""
< "%OMNIHOME%"\probes\win32\tec_db_update.sql</pre>
```

Note: To make the examples readable, the single commands are shown on multiple lines.

Forwarding events with custom variables to Tivoli Netcool OMNIbus ObjectServers

You can configure the Tivoli Log File Agent to forward custom variables in events that are sent to Tivoli Netcool/OMNIbus ObjectServers.

Before you begin

For more information about forwarding events to Tivoli Netcool OMNIbus ObjectServers, see "Forwarding events to Tivoli Netcool OMNIbus ObjectServers" on page 15.

About this task

You can configure the Log File agent to recognize and forward custom values in events that are forwarded to Tivoli Netcool OMNIbus ObjectServers. You must also configure the Tivoli Netcool/OMNIbus ObjectServer to recognize these custom attributes.

Procedure

1. Define the custom variables in the format (.fmt) file. For example, in the following format file entry TheFirst and TheSecond are custom variables added by a user.

```
FORMAT Whitespace Base
  %s %s
  TheFirst $1
  TheSecond $2
```

2. Update the Probe for Tivoli EIF tivoli eif tec.rule file. This file is under the directory path \$0MNIHOME/probes/arch. For the example, the following update is made:

```
@TECTheFirst = $TheFirst
@TECTheSecond = $TheSecond
```

3. Update the Probe for Tivoli EIF tec db update.sql file. This file is under the directory path \$0MNIHOME/probes/arch. For the example, the following updates are made:

```
alter table alerts.status add column TECTheFirst varchar(64);go
alter table alerts.status add column TECTheSecond varchar(64);go
```

- 4. Run the following command on the Tivoli Netcool/OMNIbus ObjectServer:
 - On Windows:

```
"%NCHOME%"\bin\redist\isql.exe -S server name -U username -P password
< "%OMNIHOME%"\probes\win32\tec_db_update.sql</pre>
```

• On UNIX operating systems:

```
$OMNIHOME/bin/nco sql -user username -password password -server server name
< $OMNIHOME/probes/arch/tec db update.sql
```

What to do next

At the OMNIbus event list, edit the view to list and display the custom variables. For the example described in the earlier steps, there are two fields, TheFirst, and TheSecond to add to the OMNIbus event list.

Alternatively you can also update the **Summary** field of OMNIbus to assign a custom variable. For the example described in the earlier steps, you can assign the first custom variable to the Summary field by changing the default Summary field assignment in the probes rule file from @Summary = \$msg to @Summary = \$TheFirst.

Probe for Tivoli EIF

You can use the Probe for Tivoli EIF to forward events from an external event source to the Tivoli Netcool/OMNIbus ObjectServer.

For more information, see the Netcool/OMNIbus Probe for Tivoli EIF Reference Guide.

For information about common properties and rules formats, see the IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide

Probe for Tivoli EIF properties

You use properties to specify how the probe for Tivoli EIF interacts with the Monitoring Agent for Log Files.

Note: Configure the Probe for Tivoli EIF to align with the expected event activity generated by the Monitoring Agent for Log Files.

For more detailed information about these and other properties, see the Netcool/OMNIbus Probe for Tivoli EIF Reference Guide at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIbus.doc/probes/ tivoli_eif_v11/tivoli_eif_v11/wip/reference/tveifv11_intro.html

Running the Probe for Tivoli EIF on UNIX

Use this procedure to start and stop the Probe for Tivoli EIF on UNIX systems.

Procedure

- 1. To start the probe, enter the command: \$OMNIHOME/probes/nco p tivoli eif
- 2. To stop the probe, follow these steps:
 - a. If you started it as a background process, locate the process and use the kill command to stop it.
 - b. If you started it in a console or terminal window, press Ctrl+C in the console or terminal window.

Running the Probe for Tivoli EIF on Windows

Use this procedure to start and stop the Probe for Tivoli EIF on Windows systems.

Procedure

You can start and stop the probe by using one of the following choices:

- Using a Windows command prompt
 - 1. To start the probe, open a Windows command prompt and from the directory path "%OMNIHOME%"\probes\win32 enter the command:

```
nco p tivoli eif.bat
```

- 2. To stop the probe, press Ctrl-C, or Ctrl-Break in the Windows command prompt where you started the probe.
- Install the probe to be started as a Windows service
 - 1. To install the service enter the command "%OMNIHOME%"\probes\nco p tivoli eif /INSTALL
 - 2. Go to the Windows menu option Administrative Tools > Services to start or stop the eif process.

Subnodes

Use subnodes to implement monitoring profiles that monitor multiple applications from a single agent instance.

You can accomplish the following tasks with a single agent instance by using subnodes:

- Monitor each instance of an application that is running on a system.
- Monitor different applications from a single agent instance, with different patterns applied to different logs. This type of monitoring means that logs for one application are not searched for the patterns from a different application. This type of monitoring improves efficiency.
- Monitor logs that need different duplicate detection settings (Tivoli Monitoring only) from a single agent instance.
- Organize logically separate groups of patterns, maintained by different administrators, in separate files with different write permissions without creating separate agent instances.

Creating subnodes and monitoring profiles

On the system you are monitoring, you might have multiple applications you want to monitor. To monitor all these applications from a single agent instance, you associate a profile with each application. A profile represents a single configuration and format file pairing and is implemented by using a subnode. This separation of profiles means that the logs for one application are not searched for the patterns from a different application, which improves efficiency.

To create the subnodes, you create a configuration (.conf file) and format (.fmt file) pair for each application, and place them in a specific directory location. You specify the directory location to the agent by using the Configuration file autodiscovery directory value in the configuration panel. For more information about setting configuration panel values, see "Configuring from the configuration panel" on page 6.

To organize your profiles, you can store the configuration and format file pairs in a directory tree structure under the Configuration file autodiscovery directory. Individual configuration and format file pairs must be stored in the same directory. Different configuration and format file pairs can be stored in different directories. The Log File agent checks the Configuration file autodiscovery directory location periodically, running a recursive search of the directory structure for new configuration files. The agent expects to find a format file with the same base-name as the configuration file in the same directory. For each configuration and format file pair, it finds, the agent creates a subnode by using the configuration file base-name as the profile (subnode) name. The number of subnodes is determined by the number of configuration and format file pairs. When the subnodes are created, the Log File agent begins monitoring for each configuration without further intervention.

You set configuration settings that apply globally to all subnodes when you install the agent. You set these global configuration settings by using the configuration panel or command line. You can override two of these global installation time configuration settings to use configuration file settings. The global settings that can be overridden with configuration file settings are:

- Automatically initialize UNIX syslog: You can set this value at installation time to use a configuration file value, AutoInitSyslog, that applies per subnode. To use the configuration file value you set Automatically initialize UNIX syslog to USE CONF FILE VALUE. This value is the default.
- Process Priority Class: You can set this value at installation time to use a configuration file setting, ProcessPriorityClass. To use the configuration file value you set Process Priority Class to USE_CONF_FILE_VALUE. This configuration file value does not apply per subnode as it cannot have different values in different configuration files. The limitation of not having different values is to support compatibility with earlier versions with existing configuration files. This limitation is because before the introduction of monitoring profiles that use subnodes, this value was set in the configuration file.

Defining monitoring for a number of application instances by using subnodes instead of agent instances can significantly increase the scale and performance of the Tivoli Monitoring environment.

Configuration and format files can be hosted in a central location. For more information, see "Centralized Configuration" on page 10.

Example

You have a group of WebSphere® servers for which you want to monitor the same or similar types of log files. Some servers are running a Linux operating system and others are running the AIX operating system. Each server has either a DB2® or Oracle database. Each of the operating systems and data bases has a set of log files to be monitored and patterns to be searched for in these log files.

For the purposes of the example, you want to monitor three WebSphere servers with the following log files:

```
WS server 1:
 AIX logfile
 DB2_logfile
WS server 2:
  Linux logfile
```

```
Oracle_logfile
WS_server_3:
Linux_logfile
DB2_logfile
```

To create the required subnodes for one agent instance to monitor all log files, create four pairs of configuration and format files:

```
translog_aix.conf
translog_aix.fmt
translog_db2.conf
translog_db2.fmt
translog_linux.conf
translog_linux.fmt
translog_oracle.conf
translog_oracle.fmt
```

When the file pairs are completed, you place them in the directory set in the *Configuration file* autodiscovery directory. The next time the Log File agent checks the directory, it creates a subnode for each file pair and begin monitoring for each configuration.

The configuration files in this example are deployed to servers as shown in Table 1

Table 1. Allocation of configuration files to servers

Server	Configuration files
WS_server_1	translog_aix.conf
	translog_aix.fmt
	translog_db2.conf
	translog_db2.fmt
WS_server_2	translog_linux.conf
	translog_linux.fmt
	translog_oracle.conf
	translog_oracle.fmt
WS_server_3	translog_linux.conf
	translog_linux.fmt
	translog_db2.conf
	translog_db2.fmt

This configuration means for example that WS_server_1 does not waste resources by searching for Linux or Oracle messages that never occur on this server.

Related tasks:

"Configuring from the configuration panel" on page 6

Use the configuration panel to configure the agent from the Tivoli Enterprise Portal Server graphical user interface.

Related reference:

"Configuration Files" on page 9

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

Chapter 3. Configuration file

The Tivoli Log File Agent uses a configuration file that contains configuration options and filters. You create this configuration file and configure the agent instance to use it.

The configuration file is read by the agent when it starts, and is monitored for changes to its timestamp every 60 seconds thereafter. If the timestamp of the file changes, the agent reinitializes its configuration dynamically, without requiring a restart. For more information, see "Changing the agent configuration and format files" on page 9.

The .conf file for the Log File agent accepts these options:

DupDetectionKeyAttributes

A comma-separated list of IBM Tivoli Monitoring attributes that are used to determine which events are duplicates and which are not. For more information about LogfileEvents attributes, see "LogfileEvents attribute group" on page 67. If all the named attributes are the same in two events, then those two events are considered duplicates. This option applies only to IBM Tivoli Monitoring events. For more information, see Chapter 5, "Event filtering and summarization," on page 43.

Note:

- 1. The attributes names are case-sensitive and so you must enter the names exactly as described.
- 2. If you do not provide a list of attributes, the values are defaulted to Class and Logname.

ENFORCE_STRICT_TEC_COMPATIBILITY

Set this parameter to Y to have the Log File agent treat white space characters in the same way as the Tivoli Enterprise Console Log File Adapter. This means that all white space characters in the log data are respected. For example, when you use a format such as "%s %s" to extract information from log messages, the Log File agent matches not only a literal space but also any other white space characters that are present such as tabs and carriage returns.

When this parameter is not set the default behavior of the Log File agent when it matches a Tivoli Enterprise Console style format string is to match as much of the input text as it can, while it processes the format from left-to-right.

For example, take the format string %s:%s and the input string one:two:three. The Log File agent by default assigns one.two to the first parameter (corresponding to the first %s) and it assigns three to the second parameter. The behavior of the Tivoli Enterprise Console Log File Adapter was the opposite. Setting the ENFORCE_STRICT_TEC_COMPATIBILITY parameter to Y causes the Log File agent to behave like the Tivoli Enterprise Console Log File Adapter in the way it matches formats.

Note:

- 1. This parameter applies only to format statements that use the Tivoli Enterprise Console log file adapter syntax. It does not apply to format statements that use the regular expression syntax.
- 2. Setting this parameter has a performance impact. To give greater control over the behavior and performance of matching, avoid setting this parameter, and use regular expressions instead.

EventSummaryInterval

Specifies the number of seconds during which the agent searches for duplicate events to suppress. Set to a positive integer. This option applies only to IBM Tivoli Monitoring events. For more information, see Chapter 5, "Event filtering and summarization," on page 43.

EventFloodThreshold

Specifies which events are sent when duplicate events are detected. Set to send none, send all, send_first, or a positive integer. This option applies only to IBM Tivoli Monitoring events. For more information, see Chapter 5, "Event filtering and summarization," on page 43.

EventMaxSize

Specifies in bytes, the maximum size of a generated event. If specified, this parameter is used in two places:

- 1. The parameter can be used by the agent to set the size of a buffer that is used to process events. If not set, this buffer defaults to a size of 16384 bytes. If the buffer is set too small, events are truncated and can be discarded.
- 2. The parameter can be used by the EIF sender to set the size of a buffer that is used to send events to an EIF receiver, such as the OMNIbus EIF probe. If not set, this buffer defaults to a size of 4096 bytes. If the buffer is set too small, events are discarded.

FileComparisonMode

Specifies which log files are monitored when more than one matches a wildcard pattern. The following values are available:

CompareByAllMatches

This value is the default behavior. All files that match the wildcard pattern that is specified in LogSources are monitored.

CompareByLastUpdate

Of the files that match the wildcard pattern that is specified in LogSources, the file with the most recent last update timestamp is monitored.

CompareBySize

Of the two or more files that match the file name pattern criteria, the bigger file is selected for monitoring. Do not use CompareBySize with multiple matching files that are being updated at the same time and increasing their file sizes. If the largest file is subject to frequent change, monitoring might continually restart at the beginning of the newly selected file. Instead, use CompareBySize when there is a set of matching files, but only one is active and being updated at any specific time.

CompareByCreationTime

Of the files that match the wildcard pattern that is specified in LogSources, the file with the most recent creation timestamp is monitored.

Tip: The CompareByLastUpdate, CompareBySize, and CompareByCreationTime values can all be used for rolling log files. CompareByLastUpdate is normally used for these files.

Restriction: The CompareByCreationTime value:

- Is applicable only to Windows operating systems as UNIX and Linux operating systems do not store a true creation time for files.
- Is not supported for remote files that you monitor by using the Secure Shell (SSH) File Transfer Protocol.

FQDomain

Specifies how and if the agent sets a domain name.

- If set to yes, the agent determines the system domain name itself.
- If set to no, the agent does not set a domain name. The fqhostname attribute is assigned a blank string.
- If set so that it does not contain a yes or no value, the domain name is accepted as the value and it is appended to the host name.

For more information, see Chapter 4, "Format file," on page 33.

IncludeEIFEventAttr

The agent includes a large attribute that is called *EIFEvent*, which is a representation of the event that would be sent through the Event Integration Facility if that feature is enabled. The information that is contained in the *EIFEvent* attribute can also be found in other attributes. Its large size made it problematic, thus it was disabled in version 6.2.3 and no value is shown if viewed on the Tivoli Enterprise Portal. Setting this value to y, re-enables the EIFEvent attribute in Tivoli Monitoring.

Note: Using this attribute might cause situations to fail if you have large events. A large event in this context is an event where the total number of bytes necessary to contain all of the values, for all attributes, and their names, results in a string longer than 3600 bytes.

LognameIsBasename

When set to y, the value of the Logname attribute in IBM Tivoli Monitoring is the base name of the log file in which the event was found. This option applies only to IBM Tivoli Monitoring events. The path is removed. For example, /data/logs/mylog.log becomes mylog.log. If this value is set to n, then you get the full path. However, because the attribute is limited to 64 characters, setting it to n means that the name is truncated if it is longer. For this reason, the default value is y. To see the full path name in a longer attribute, you can specify it in the mappings section of a format in the .fmt file, for example, filename FILENAME CustomSlot1. The mapping completes the slot named filename with the full path of the file in which the event was found and maps it into CustomSlot1 that is 256 characters long.

LogSources

Specifies the text log files to poll for messages. The complete path to each file must be specified, and file names must be separated by commas. Within each file name, you can also use an asterisk (*) to represent any sequence of characters, or a question mark (?) to represent any single character. For example, mylog* results in polling all log files whose names begin with mylog, whereas mylog??? results in polling all log files whose names consist of mylog followed by exactly three characters. These wildcard characters are supported only within the file name; the path must be explicitly specified.

If you want to use regular expressions or pattern matching in the path, see the RegexLogSources description.

A log file source is not required to exist when the agent is started; the log file is polled when it is created.

NewFilePollInterval

Specifies the frequency, in seconds, that the agent checks for new files to monitor. For example, if a file name specified by the *LogSources* or *RegexLogSources* configuration file settings does not yet exist, when the agent starts it checks again for the files existence after this interval.

NumEventsToCatchUp

Specifies the event in the log that the agent starts with. This option provides some flexibility if the source that is being monitored is new or the agent is stopped for an extended time. Valid values are as follows:

Note: For text files, values 0 and -1 apply. For Windows event log, values 0, -1, and n apply.

- 6 Start with the next event in the logs. This value is the default.
- -1 When set to -1, the agent saves its place in the file that is being monitored. It saves its place so that when the agent is stopped and later restarted, it can process any events that are written to the log while it was stopped. The agent otherwise ignores events that arrived while it was stopped and restarts from the end of the file. This setting does not apply to pipes, or syslog monitoring on UNIX and Linux.
- **n** Set to a positive integer. Starts with the *nth* event from the most current event in the logs;

that is, start n events back from the most current event in the logs. If n is greater than the number of events that are available, all the events that are available are processed.

Note: You can use the n value only for Windows Event Log. The n value is ignored when UseNewEventLogAPI is set to y.

PollInterval

Specifies the frequency, in seconds, to poll each log file that is listed in the LogSources option for new messages. The default value is 5 seconds

If you upgraded a Windows event log adapter from a previous release and you have a value that is set for PollingInterval in the Windows registry, you must specify the PollInterval option in the agent configuration file with the same value used in the Windows registry. This rule applies only if you are replacing a Tivoli Enterprise Console log file agent that had values in the registry.

ProcessPriorityClass

Specifies the process priority for the agent. You can adjust this value to improve system performance if the agent processes large volumes of events and is using too many processor resources. The possible values are:

- A Very low priority
- B Low priority
- C Typical priority
- D Above typical priority
- E High priority
- F Very high priority
- USE CONF FILE VALUE Use the value that is specified in the configuration file (This value is the default)

Note: The value of this setting is global. If you implement monitoring profiles by using subnodes this value applies to all monitoring profiles. Its value cannot be different in different configuration files for the same agent.

RegexLogSources

Specifies the text log files to poll for messages. It differs from the LogSources option in that regular expression meta characters can be used in the base name portion of the file name and at most one subdirectory of the file name. This difference provides greater flexibility to describe multiple files to monitor in multiple directories than the LogSources option.

For example, specifying /var/log/mylog* for the LogSources statement is identical to using the dot (.) meta character followed by an asterisk (*) meta character to form /var/log/mylog.* in the RegexLogSources statement. This type of qualifier results in polling all log files in the /var/log directory whose base names begin with mylog and are followed by zero or more characters. A /var/log/mylog.+ qualifier results in polling all log files in the /var/log directory whose names begin with mylog and are followed by one or more characters.

Similar to LogSources, the complete path to each file must be specified and the file names must be separated by commas. However, the comma is also a valid character inside a regular expression. In order to distinguish between a comma that is used as part of a regular expression and one used to separate file names, commas that are used as part of a regular expression must be escaped with the backslash (\) character.

For example, if you want to search for logs that match either of the following regular expressions: /logs/.*\.log and /other/logs/[a-z] $\{0,3\}$ \.log, you must escape the comma in the $\{0,3\}$ clause of the second expression, so that the agent does not mistake it for the beginning of a new expression: RegexLogSources=/logs/.*\.log,/other/logs/[a-z]{0\,3}\.log

If meta characters are used in the path name, the meta characters can be used in only one subdirectory of the path. For example, you can specify /var/log/[0-9\.]*/mylog.* to have meta characters in one subdirectory. The [0-9\.]* results in matching any subdirectory of /var/log that consists solely of numbers and dots (.). The mylog.* results in matching any file names in those/var/log subdirectories that begin with mylog and are followed by zero or more characters.

Because some operating systems use the backslash (\) as a directory separator it can be confused with a regular expression escape meta character. Because of this confusion forward slashes must always be used to indicate directories. For example, Windows files that are specified as C:\temp\mylog.* might mean the \t is a shorthand tab character. Therefore, always use forward slashes (/) on all operating systems for directory separators. The C:/temp/mylog.* example represents all files in the C:/temp directory that start with mylog.

If more than one subdirectory contains meta characters, a trace message is also issued. For example, c:/[0-9\.]*/temp.files/mylog.* has two subdirectories with meta characters. [0-9\.]* is the first subdirectory with meta characters and temp.files is the second subdirectory that used a dot (.) meta character. In this case, the agent assumes that the first subdirectory with the meta character is used and the subsequent directories with meta characters are ignored. For more information about troubleshooting issues with regular expressions, see "Agent troubleshooting" on page 115.

SubnodeName

A string value that can be used to override the default name that is assigned to a monitoring profile subnode. By default the subnode name that is assigned to a monitoring profile corresponds to the base name of the configuration file that is used for that profile. Using this setting a different subnode name can be assigned.

SubnodeDescription

A string value that can be used to assign a value to the *Subnode Description* attribute of the *LFAProfiles* workspace.

UnmatchLog

Specifies a file to log discarded events that cannot be parsed into an event class by the agent. The discarded events can then be analyzed to determine whether modifications are needed to the agent format file. Events that match a pattern that uses *DISCARD* do not appear in the unmatch log because they did match a pattern.

This option is generally used in a test environment to validate the filters in the format file. This option fills up your file system if you leave it on for extended periods

Options for remote log file monitoring by using SSH

Other than **SshHostList**, which is a list, all options can have only one value, which is applied to all remote hosts specified in **SshHostList**.

Only text log files are supported. AIX error report, syslog, and Windows event log are not supported.

Tip: You can set up syslog to write its output to a text log file and then remotely monitor that text file with log file agent.

SshAuthType

Must be set to either *PASSWORD* or *PUBLICKEY*. If set to *PASSWORD*, the value of **SshPassword** is treated as the password to be used for SSH authentication with all remote systems. If set to *PUBLICKEY*, the value of **SshPassword** is treated as the pass phrase that controls access to the private key file. If set to *PUBLICKEY*, **SshPrivKeyfile** and **SshPubKeyfile** must also be specified.

SshHostList

A comma-separated list of remote hosts to monitor. All log files that are specified in the **LogSources** or **RegexLogSources** statements are monitored on each host that is listed here. If one of the host names specified is *localhost*, the agent monitors the same set of files directly on the local system. When you specify *localhost*, SSH is not used to access the files on the local system, the log files are read directly.

SshPassword

When the value of **SshAuthType** is *PASSWORD*, this value is the account password of the user that is specified in SshUserid. You can supply the account password in clear text, or you can supply a password that is encrypted with the IBM Tivoli Monitoring itmpwdsnmp command. For more information about how to encrypt a password by using the itmpwdsnmp command, see "Encrypting a password or pass phrase" on page 13.

When the value of **SshAuthType** is *PUBLICKEY*, this value is the pass phrase that decrypts the private key that is specified by the SshPrivKeyfile parameter. You can supply the pass phrase in clear text, or you can supply a pass phrase that is encrypted with the IBM Tivoli Monitoring itmpwdsnmp command. For more information about how to encrypt a password by using the itmpwdsnmp command, see "Encrypting a password or pass phrase" on page 13.

Note: If the value of **SshAuthType** is *PUBLICKEY*, and you configured SSH not to require a pass phrase, SshPassword must be set to null. To set SshPassword to null, the entry in the configuration file is:

SshPassword=

SshPort

A TCP port to connect to for SSH. If not set, defaults to 22.

SshPrivKeyfile

If **SshAuthType** is set to *PUBLICKEY*, this value must be the full path to the file that contains the private key of the user that is specified in SshUserid and SshPubKeyfile must also be set. If **SshAuthType** is not set to *PUBLICKEY*, this value is not required and is ignored.

SshPubKeyfile

If **SshAuthType** is set to *PUBLICKEY*, this value must be the full path to the file that contains the public key of the user that is specified in **SshUserid** and **SshPrivKeyfile** must also be set. If **SshAuthType** is not set to *PUBLICKEY*, this value is not required and is ignored.

SshUserid

The user name on the remote systems which the agent uses for SSH authentication.

The following option is supported on UNIX and Linux only:

AutoInitSyslog

If this option is set to Yes, the agent automatically configures the syslog facility to write a standard set of events to a pipe that the agent monitors. By enabling this setting, you can monitor syslog events without maintaining and rolling-over log files. If not set in the configuration file, it is the same as being set to No.

Restriction: This option is not supported for remote log file monitoring.

The following options are supported on Windows only:

NTEventLogMaxReadBytes

If you are using the older NT Event Log interface (UseNewEventLogAPI is not set to y) to read Event Log data on a Windows system, the agent reads up to this number of bytes each time it checks the Event Log for new data. Setting the value to 0 causes the agent to attempt to read all new data, as it did in earlier releases. This activity can occupy the agent for a considerable amount of time on a system with many events. The default value is 655360. When set, the agent might not stop at exactly the value specified, but rather at the nearest multiple of an internal buffer size to this value.

PreFilter

Specifies how events in a Windows Event Log are filtered before agent processing. PreFilter statements are used by PreFilterMode when determining which events are sent from an event log to the agent. An event matches a PreFilter statement when each attribute=value specification in the PreFilter statement matches an event in the event log. A PreFilter statement must contain at

least the log specification and can contain up to three further specifications, which are all optional: event ID, event type, and event source. The order of the attributes in the statement does not matter.

The basic format of the PreFilter statement is as follows:

PreFilter:Log=log name;EventId=value; EventType=value;Source=value;

You can specify multiple values for each attribute by separating each with a comma.

Each PreFilter statement must be on a single line.

PreFilter is not mandatory. All Windows log events are sent to the agent if prefilters are not specified and PreFilterMode=OUT.

PreFilterMode

This option applies only to Windows Event Log. Specifies whether Windows log events that match a PreFilter statement are sent (PreFilterMode=IN) or ignored (PreFilterMode=OUT). Valid values are IN, in, OUT, or out. The default value is OUT.

PreFilterMode is optional; if PreFilterMode is not specified, only events that do not match any PreFilter statements are sent to the agent.

Note: If you set PreFilterMode=IN, you must also define the PreFilter statements.

SpaceReplacement

Set to TRUE by default for Windows Event Log (Windows Server 2008 only) but not for previous versions of event log. When SpaceReplacement is TRUE, any spaces in the security ID, subsource, Level, and keywords fields of the event log messages are replaced with underscores (_). When SpaceReplacement is FALSE, any spaces in the security ID, subsource, Level, and keywords fields of the event log messages remain unchanged. For more information about this option, see Appendix B, "Windows 2008 event log," on page 137.

UseNewEventLogAPI

When set to y on Windows systems, uses the new Windows Event Log interface for event logs. The option is supported only on Windows 2008 and later. The option is needed to access many of the new event logs that debuted in Windows 2008 and the applications that run on it. The option is ignored on earlier versions of Windows and on UNIX and Linux. For more information about this option, see Appendix B, "Windows 2008 event log," on page 137.

WINEVENTLOGS

Controls which Windows event logs are monitored.

The WINEVENTLOGS statement is a comma-delimited list with no spaces. For more information, see Appendix B, "Windows 2008 event log," on page 137.

Note: Any carriage returns, tabs, or newlines in Windows events are replaced by spaces.

The following option is supported on AIX only:

AIXErrptCmd

An **errpt** (error report) command string can be supplied here, which the agent runs. The command output is fed into the stream of log data that is being monitored.

For example, the following command causes the agent to search for the mmddhhmmyy string and replace it with the actual date and time on startup. Only the first occurrence of the string is replaced.

AIXErrptCmd=errpt -c -smmddhhmmyy

Although you can supply your own errpt command, the Log File agent enforces the same constraints on this command that the Monitoring Agent for UNIX Logs does. In particular, you must use the -c (concurrent mode) option so that the command runs continuously. You cannot use the -t option or the following options that result in detailed output: -a, -A, or -g. The data

stream is the standard output from the errpt command, so regular expressions in the fmt file must be written to match. For example, the data output might be:

```
IDENTIFIER TIMESTAMP T C RESOURCE NAME DESCRIPTION
F7FA22C9 0723182911 I 0 SYSJ2
                                    UNABLE TO ALLOCATE SPACE IN FILE SYSTEM
2B4F5CAB
          1006152710 U U ffdc
                                    UNDETERMINED ERROR
2B4F5CAB
          1006152610 U U ffdc
                                    UNDETERMINED ERROR
```

A sample format that picks up the data rows, but not the header, would be:

```
REGEX GenericErrpt
^([A-F0-9]{8}) +([0-9]{10}) ([A-Z]) ([A-Z]) (\S+) +(.*)$
Identifier $1 CustomSlot1
Timestamp $2 CustomSlot2
          $3 CustomSlot3
С
          $4 CustomSlot4
Resource $5 CustomSlot5
          $6
msg
END
```

see also the Agent Builder User's Guide under Monitoring an AIX Binary Log.

The following options apply only where events are being forwarded to EIF:

BufferEvents

Specifies how event buffering is enabled. The possible values are:

- · YES Stores events in the file that is specified by the BufEvtPath option (This value is the default).
- MEMORY_ONLY Buffers events in memory.
- NO Does not store or buffer events.

BufEvtPath

Specifies the full path name of the agent cache file. If this path is not ecified the default is:

- UNIX systems: For more information/etc/Tivoli/tec/cache
- Windows systems: \etc\Tivoli\tec\cache

Note: If events are being forwarded to more than one server, a *BufEvtPath* value must be specified for each forwarding channel. An index number is appended to the BufEvtPath name for each additional entry. For example, use BufEvtPath1 to indicate the path name of the agent cache file for forwarding to the first additional server. The value that is set in each BufEvtPath must be unique.

BufEvtMaxSize

Specifies the maximum size, in KB, of the agent cache file. The default value is 64. The cache file stores events on disk when the BufferEvents option is set to Yes. The minimum size for the file is 8 KB. File sizes specified less that this level are ignored, and 8 KB is used. There is no upper limit to the value you specify for the maximum file size.

Note: If the cache file exists, you must delete the file for option changes to take effect.

NO UTF8 CONVERSION

Specifies whether Tivoli Event Integration Facility encodes event data in UTF-8. When this option is set to YES, Tivoli Event Integration Facility does not encode event data in UTF-8. The data is assumed to already be in UTF-8 encoding when passed to Tivoli Event Integration Facility. A prefix is however added to the flag to indicate that the data is in UTF-8 encoding (if the flag does not exist at the beginning of the event data). The default value is NO.

MaxEventQueueDepth

This value indicates the maximum number of events that can be queued for forwarding. When

the limit is reached each new event that is placed on the queue bumps the oldest event from the queue. If not specified the default value is 1000. This setting applies to all forwarding channels if NumAdditionalServers is used.

NumAdditionalServers

This entry is required if you want to forward events to more than one Netcool OMNIbus ObjectServer. Its value is used to indicate the number of servers that events are forwarded to. Valid values are 1-8.

ServerLocation

Specifies the name of the host on which the event server is installed. Specify host_name or IP_address. Use the dotted format for IP_address. You can specify failover values for the server locations if you want to, such as: ServerLocation1=2.3.4.5,2.3.4.6., If you specify failover values for ServerLocation you must also specify an additional ServerPort value for each ServerLocation.

Note: If events are being forwarded to more than one server, a ServerLocation value must be specified for each server. An index number is appended to the ServerLocation name for each additional entry. For example, use ServerLocation1 to specify the name of the host on which the first additional server is installed.

ServerPort

Specifies the port number on which the EIF receiver listens for events. The ServerPort option can contain up to eight values, which are separated by commas. If failover values are specified for ServerLocation you must set an equivalent ServerPort value. The ServerPort is not used when the *TransportList* option is specified.

Note: If events are being forwarded to more than one server, a ServerPort value must be specified for each server. An index number is appended to the ServerPort name for each additional entry. For example, use ServerPort1 to specify the port number on which the EIF receiver listens for events for the first additional server.

TransportList

Specifies the user-supplied names of the transport mechanisms, which are separated by commas. When a transport mechanism fails for sender applications, the API uses the following transport mechanisms in the order that is specified in the list. For receiving applications, the API creates and uses all the transport mechanisms. The transport type and channel for each type_namemust be specified by using the Type and Channels keywords:

type_name**Type**

Specifies the transport type for the transport mechanism that is specified by the *TransportList* option. SOCKET is the only supported transport type.

The server and port for each channel_name are specified by the ServerLocation and ServerPort options.

type_nameChannels

channel namePort

Specifies the port number on which the transport mechanisms server listens for the specified channel (set by the Channel option). When this keyword is set to zero, the portmapper is used. This keyword is required.

channel namePortMapper

Enables the portmapper for the specified channel.

channel namePortMapperName

If the portmapper is enabled, specifies the name of the portmapper.

channel_namePortMapperNumber

Specifies the ID registered by the remote procedure call.

channel_namePortMapperVersion

If the portmapper is enabled, specifies the version of the portmapper

channel_nameServerLocation

Specifies the name of the event server and region on which the server for transport mechanisms is located for the specified channel. The channel is set by the Channel option. This keyword is required.

Unsupported configuration file options:

The following .conf file options that are supported by the Tivoli Enterprise Console Log File Adapter are ignored by the Log File agent.

- BufferMaxSize
- HostnameIsAdapterHost
- LanguageID
- ProcessDisablePriorityBoost
- NewLogBasedOn
- PingTimeout
- NumberOfPingCalls

Note: Other than the previously listed configuration file options, the Log File agent supports all of the other configuration file options that are documented in the IBM Tivoli Enterprise Console Adapters Guide.

The configuration file accepts generic EIF options when used directly with OMNIbus. These options operate only over an EIF connection to OMNIbus. They do not affect events that are sent to IBM Tivoli Monitoring. For more information about these EIF options, see EIF keywords.

Related concepts:

Chapter 5, "Event filtering and summarization," on page 43

Use the event filtering and summarization options that you set in the configuration (.conf) file to control how duplicate events are handled by the Tivoli Log File Agent.

Related reference:

"LogfileEvents attribute group" on page 67

Events matching configured formats in monitored log files.

"Changing the agent configuration and format files" on page 9

The Tivoli Log File Agent reads its configuration (.conf) and format (.fmt) files when it starts, and monitors their timestamp every 60 seconds thereafter.

"Configuration Files" on page 9

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

Chapter 4. Format file

The Tivoli Log File Agent extracts information from system log messages and then matches different log messages to event classes. A format file serves as a lookup file for matching log messages to event classes, telling the event class what to read, what to match, and how to format the data.

When the format file is used as a lookup file, all format specifications in the file are compared from the beginning to the end of the file. When two classes match or there are multiple matching classes for a message, the first expression from the end that matches is used. If no match is found, the event is discarded. A discarded event is written to the unmatch log if it is defined in the .conf file.

In this chapter, the regular expression syntax that you use to create patterns to match log messages and events is described. Regular expression-filtering support is provided by using the International Components for Unicode (ICU) libraries to check whether the attribute value examined matches the specified pattern.

For more information about using regular expressions, see http://userguide.icu-project.org/strings/regexp.

Note: For compatibility purposes, the syntax used by the Tivoli Enterprise Console log file adapter is still supported. Any new formats added to existing files are also supported. However, for any new format files that you create, use the syntax described here. If you require more information about the syntax used by the Tivoli Enterprise Console log file adapter, see Format file reference

Related reference:

"Configuration Files" on page 9

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

Format file specifications

The format file describes the patterns that the agent looks for to match events in the monitored logs. The format file consists one or more format specifications.

You can change the format file while an agent instance is running. The file is read by the agent when it starts, and is monitored for changes to its timestamp every 60 seconds thereafter. If the timestamp of the file changes, the agent reinitializes its configuration dynamically, without requiring a restart. For more information, see "Changing the agent configuration and format files" on page 9.

To create new patterns to match an event, use the new regular expression syntax that consists of the following parts:

- · Format header
- · Regular expression
- · Slot mappings
- End statement

The format header contains the keyword REGEX, which informs the agent that you are using a regular expression to match the pattern in the monitored log.

You assign this regular expression to an event class as shown in the following example: REGEX REExample

If you use the special predefined event class *DISCARD* as your event class, any log records matching the associated pattern are discarded, and no events are generated for them. For example: REGEX *DISCARD*

As a pattern matched, nothing is written to the unmatch log. The log file status records matched include these discarded events.

Note: You can assign multiple event definitions to either the same event class or to different event classes. The class name is arbitrary and you can use it to indicate the type of event or to group events in various ways.

After the format header, the format content consists of a regular expression on the first line, followed by mappings. Each mapping is shown on a separate line and these mappings are described in the following example.

All lines that match the regular expressions are selected and sent to the monitoring server as events. The regular expression contains sub expressions. You can use the sub expressions to match specific parts of these lines that match, to a variable called a 'slot' in the Event Integration Facility. You can optionally map these slots to specific attributes in IBM Tivoli Monitoring.

The monitoring log shown here contains three lines that you might want to monitor:

Error: disk failure Error: out of memory WARNING: incorrect login

For example, you generate an event for a specific error, such as the lines that begin with Error: and ignore the line that begins with Warning:. The regular expression must match the lines that begin with Error: and also include a sub expression. The sub expression is denoted by parentheses and it must match only the input text that you want to assign to the msg slot. An example is shown in the following format definition:

REGEX REExample Error: (.*) msg \$1 **END**

Note: This example shows a simple regular expression with only one sub expression.

Based on this format specification, and the preceding set of log data, the agent generates two events. Both events are assigned the REEXample event class. In the first event, the disk failure value is assigned to the msg slot. Also, in the second event, the out of memory value is assigned to the msg slot. As the Warning line did not match the regular expression, it is ignored and no event is generated.

When you assign the value of \$1 to the msg slot, you assign it the value of the first sub expression.

If you have log text that contains the following errors:

```
Error: disk failure on device /dev/sd0: bad sector
Error: disk failure on device /dev/sd1: temperature out of range
```

you might want to assign these error messages to their own event class so that you are informed immediately if there is a disk failure. You can include a description of the disk on which the error occurred and more specifically the disk error in the event.

The following regular expression contains two sub expressions that identify this information:

```
REGEX DiskFailure
Error: disk failure on device (/dev/sd[0-9]):(.*)
device $1 CustomSlot1
msg $2
END
```

You assign these two sub expressions to event slots. The two events that are generated contain the following values:

```
"device=/dev/sd0" and "msg=bad sector"
"device=/dev/sd1" and "msg=temperature out of range"
```

If you use EIF to generate the first event, it displays as shown in the following example: DiskError;device='/dev/sd0';msg='bad sector';END

If the event is sent to IBM Tivoli Monitoring, the slot named *msg* is assigned to the IBM Tivoli Monitoring attribute of the same name. But there is no pre-defined IBM Tivoli Monitoring attribute for the slot that we called *device*.

If you need to see the value assigned to *device* directly on the Tivoli Enterprise Portal, or write situations against it, you must assign it to an IBM Tivoli Monitoring attribute.

The Tivoli Log File agent includes the following 13 predefined attributes:

• 10 string type attributes ranging from

CustomSlot1 to CustomSlot10.

• Three integer type attributes ranging from

CustomInteger1 to CustomInteger3.

Using these attribute names in the format file populates IBM Tivoli Monitoring attributes of the same name. Using these attributes does not affect the content of the EIF event sent directly to OMNIbus.

Note: The CustomSlot and CustomInteger attribute names are case-sensitive and so you must enter the names exactly as shown.

You assign a slot from the event definition to one of these custom IBM Tivoli Monitoring attributes in the format file.

You assign the *device* slot to the IBM Tivoli Monitoring string type attribute called *CustomSlot1* as shown in the following example:

```
REGEX DiskFailure
Error: disk failure on device (/dev/sd[0-9]):(.*)
device $1 CustomSlot1
msg $2
END
```

When the event displays in IBM Tivoli Monitoring, the value assigned to the *device* slot is assigned to the IBM Tivoli Monitoring attribute called CustomSlot1. You view this value on the Tivoli Enterprise Portal or use it to define situations. You can assign any slot in the event definition to any of the 10 custom IBM Tivoli Monitoring attributes in the same manner, by putting "CustomSlot<n>", where <n> is a number from 1 to 10, next to the slot definition.

In this example, the first sub expression is defined specifically as (/dev/sd[0-9]), but the second sub expression is defined generally as (.*). In defining the regular expression as specifically as possible, you improve performance. Therefore, if you enter a search for an error on a device that does not match the specific error message defined here, the search procedure stops immediately when the error is not found. Time is not wasted looking for a match.

The *END* keyword completes the format specification. The format header, regular expression, and the *END* keyword must each begin on a new line, as shown in the following example:

```
REGEX REExample
Error:
msg $1
END <EOL>
<EOF>
```

Note: For the last format in the file, you must insert a newline after the **END** keyword as shown in the example. Otherwise, you get a parsing error.

The custom integer attributes *CustomInteger1* to *CustomInteger3* are 64-bit integer attributes. You can use them in the same manner as the string type CustomSlot attributes. You can use these attributes to map individual slots, or sub expressions, from the log file to individual IBM Tivoli Monitoring attributes. Because these attributes are numeric, you can use arithmetic comparisons on them, such as < and >, which is not possible with the string attributes.

Note: Although these values are evaluated as integers by the IBM Tivoli Monitoring infrastructure, for EIF purposes, and within the format file, they are still treated as strings. So for example, to use an integer slot in a PRINTF statement, you still identify it with "%s", not "%d".

The following example illustrates the use of a custom integer attribute. Suppose that a periodic UNIX syslog message is received that reports the percentage of a file system that is free, such as the following hypothetical log record:

```
Oct 24 11:05:10 jimmy fschecker[2165]: Filesystem /usr is 97% full.
```

You can use the following statement in the format file to check for the percentage of the file system that is free:

```
REGEX FileSystemUsage
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2}) (.*?) (.*?):
Filesystem (.*?) is ([0-9]+)% full\.$
Month $1 CustomSlot1
Date
       $2 CustomSlot2
     $3 CustomSlot3
Time
Host $4 CustomSlot4
Service $5 CustomSlot5
Filesystem
             $6 CustomSlot6
PctFull
            $7 CustomInteger1
            PRINTF("%s: %s% full", Filesystem, PctFull)
END
```

Note: In the preceding statement, everything between the ^ and \$ symbols on the second and third lines must be on a single line.

Because you might have other events that put values in *CustomInteger1*, you can avoid confusing the different event types by using the value of the *Class* attribute to limit its effect to the correct type of events. For example, the following situation formula causes the situation to fire only when an event of the *FileSystemUsage* event class has a value greater than or equal to 95 in *CustomInteger1*:

```
( Class == 'FileSystemUsage' AND CustomInteger1 >= 95)
```

A different event can then use *CustomInteger1* for a different purpose and not trigger this situation accidentally.

In summary, you can now write a situation in IBM Tivoli Monitoring that uses arithmetic operators on the CustomInteger attributes, which is not possible with the CustomSlots attributes.

Note: If you map non-integer data to the CustomInteger attributes the resulting value might be zero or some unexpected value.

Related reference:

"Log File Status attribute group" on page 63

The Log File Status attribute group contains information that reflects the status of log files this agent is monitoring.

"Changing the agent configuration and format files" on page 9

The Tivoli Log File Agent reads its configuration (.conf) and format (.fmt) files when it starts, and monitors their timestamp every 60 seconds thereafter.

"Configuration Files" on page 9

The configuration (.conf and .fmt) files tell the agent what log files to monitor, where to find them and what formatting to use to read them.

Inheritance

A format file uses inheritance to derive slot definitions from a previously defined format specification.

Use the FOLLOWS relationship to build specific format specifications from generic format specifications by using inheritance.

First, you define a base class and call it DiskFailure, for example, as shown here:

```
Disk Failure on device (.*)
device $1 CustomSlot1
END
```

This regular expression matches the Disk Failure on device/dev/sd0 errors in the monitoring log so that the /dev/sd0 value is assigned to the *device* slot.

However, you can also see an extended version of this error message reported in the monitoring log.

For example, you might see a Disk Failure on device /dev/sd0, error code: 13 error message.

This error message is matched to a slot as shown in the following example:

```
REGEX DiskFailureError FOLLOWS DiskFailure
Disk Failure on device (.*), error code: ([0-9]*)
errcode $2 CustomSlot2
END
```

Now, the event includes the *device* slot and the *errcode* slot. As the DiskFailure event class defined a slot for the device name already, you allow the subclass to inherit that slot, and this inheritance saves you from declaring it a second time. The slot is defined as \$1 so the first sub expression in the regular expression is assigned to that slot.

However, the DiskFailureError class also defines a second sub expression. You can assign this sub expression to a new slot called errcode and define it as \$2 to refer to the second sub expression in the regular expression. This type of assignment is shown in the previous example that displays the log text.

The event now contains the device slot that is assigned the /dev/sd0 value and the errcode slot that is assigned a value of 13. CustomSlot1 is assigned the device, and CustomSlot2 is assigned the error code.

IBM Tivoli Monitoring custom attribute mappings are also inherited. For more information about IBM Tivoli Monitoring custom attribute mappings, see "Format file specifications" on page 33.

Multi-line

Use the multi-line syntax to match records that span more than one line to patterns in the log that you are monitoring.

Specify the \n newline character as part of the regular expression to indicate where the line breaks occur in the monitoring log. See this type of syntax in the following example:

```
REGEX REMultiLine
Line1:(.*)\nLine2(.*)
msg $1
second_msg $2
END
```

Note: On Windows, you specify a \r\n carriage-return and newline combination.

If the following error messages are reported in the log text, the REMultiLine event is created:

Line1: An error occurred Line2: The error was "disk error"

The msg slot is assigned the value of An error occurred and the second_msg slot is assigned the value of The error was "disk error".

Mappings

The Tivoli Log File Agent uses mappings to determine the event class for a system log message. The agent determines the event class by matching the message to a pattern in the format file.

The agent translates log messages in to event class instances that contain attribute name=value pairs. The event is then sent to the event server.

The agent determines the event class for a system log message at the source. The agent determines the event class by matching a system log message to a pattern in the format file. After you use this matching procedure to determine a class, you must assign values to the attributes.

Attribute values come from various sources, such as:

- Default values provided by the agent.
- Log text that matches specific sub expressions in regular expressions.

A map statement is included in the format file and it consists of the following syntax:

name value CustomSlotn

Here, you specify any identifier to describe the name of a slot (also known as a variable, attribute, or value identifier). Then, you specify a value to assign to this slot by applying any of the values that are described in "Value specifiers" on page 39.

Use custom slots to view data in the Tivoli Enterprise Portal and to define situations. When creating situations, all custom slot values are strings. Custom slots are also needed to get duplicate detection to work because you must identify the slots used to determine duplicates. For more information about filtering events, see Chapter 5, "Event filtering and summarization," on page 43. msg is a special slot name, with its own attribute in the event table. You do not need to use a custom slot for the msg.

You can limit the scope of a slot so that it exists only within the format definition. When you define the slot, you precede the slot name with a dash, for example:

-name value

Any slot that you define in this way is not included in the final event. However, you can reference the slot elsewhere in the format definition, specifically within a PRINTF statement. In the REGenericSyslog example that follows, the service slot is not included in the event that you generate but you can reference it in the PRINTF statement. It retains the same value that was applied to the original slot when it was defined without the dash. Using this procedure you can use temporary variables from the format

definition that are not included in the final event. For example, you can define an event class, REGenericSyslog, to match generic UNIX syslog events in the following way:

```
REGEX REGenericSyslog
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2}) (.*?) (.*?): (.*)$
month $1
date $2
time $3
host $4
-service $5
msg $6
syslog_msg PRINTF("service %s reports %s", service, msg)
```

Value specifiers

This section describes how the mappings in a format specification assign values to attributes.

The mapping part of a format specification consists of the following types of value specifiers:

- \$i
- · String constant
- PRINTF statement

\$i

The i indicates the position of a sub expression in a format string. Each sub expression is numbered from 1 to the maximum number of sub expressions in the format string.

The value of a \$i value specifier (also known as a variable, slot, or attribute) is the portion of the system log message that is matched by the corresponding sub expression.

In the following example, the log agent translates any log message from the UNIX syslog facility into a syslog event with values assigned to it:

```
REGEX REGenericSyslog
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2})
(.*?) (.*?): (.*)$
month $1
date $2
time $3
host $4
service $5
msg $6
END
```

Each sub expression numbered from \$1 to \$6 matches an item in parentheses in the regular expression.

```
Therefore, the following syslog event:
```

```
Apr 6 10:03:20 jimmy syslogd 1.4.1: restart.
```

is assigned the following values:

```
month=Apr
date=6
time=10:03:20
host=jimmy
service=syslogd 1.4.1
msg=restart.
```

For example, in the syslog event, the 10:03:20 value matches the third item in parentheses in the regular expression, so the value is assigned to the \$3 time value. Similarly, the jimmy value matches the fourth item in parentheses in the regular expression, so the value is assigned to the \$4 host value.

string constant

The string constant declares that the value of the attribute is the specified string. If the attribute value is a single constant without any spaces, you specify it without surrounding double quotation marks (" ") as shown in the following example:

severity WARNING

Otherwise, if there are spaces in the attribute value, double quotation marks must be used as shown in the following example:

component "Web Server"

PRINTF statement

The PRINTF statement creates more complex attribute values from other attribute values. The PRINTF statement consists of the keyword **PRINTF** followed by a printf() C-style format string and one or more attribute names.

The format string supports only the %s component specifier. The values of the attributes that are used in the PRINTF statement must be derived from either a \$i value specification or a constant string value specification (you cannot derive them from another PRINTF statement).

Use the value of the argument attributes to compose a new constant string according to the format string. This new constant string becomes the value of the attribute.

Based on the previous example where you defined the REGenericSyslog base class, and the *service* and *msg* slots, you can define an attribute called *syslog_msg* by using the **PRINTF** keyword.

syslog msg PRINTF("service %s reports %s", service, msg)

If the following log message is reported:

Apr 6 10:03:20 jimmy syslogd 1.4.1: restart.

A new constant string is composed containing the attribute values from the format string as shown here:

syslog_msg="service syslogd 1.4.1 reports restart."

Keywords

Use Keywords to assign values in the format file that expand at run time.

The following keywords expand at run time:

- DEFAULT
- FILENAME
- LABEL
- REGEX

DEFAULT

Use the **DEFAULT** keyword to assign a DEFAULT value to a specific slot or attribute. The Tivoli Log File agent assigns an internal default value to slots that are described in the following table:

Table 2. Slots and the DEFAULT value

Slots	Description
hostname	<i>hostname</i> is the short host name of the system where the agent is running. It does not include the domain name of the system.
origin	<i>origin</i> is the IP address of the system where the agent is running.

Table 2. Slots and the DEFAULT value (continued)

Slots	Description
fqhostname	<i>fqhostname</i> is the fully qualified host name of the system where the agent is running. It includes the domain name of the system.
RemoteHost	When an event originates on the local system, this attribute is empty. If an event originates on a remote system, <i>RemoteHost</i> contains a string of the form <i>user@host:port</i> , indicating the remote host name on which the event occurred, and the user and port on that host that are used to connect.

The value assigned to *fqhostname* is influenced by the following FQDomain (optional) settings in the .conf file:

- · If you set FQDomain to a 'yes' value, then the agent determines the system domain name itself.
- If you do not set a value for FQDomain or if you set a 'no' value, then the agent does not set a domain name. The *fqhostname* attribute is assigned a blank string.
- Finally if you set FQDomain so that it does not contain a 'yes' or 'no' value, the domain name is accepted as the value and it is appended to the host name.

In the following example, the format definition contains three attributes or slots:

- hostname DEFAULT
- origin DEFAULT
- fqhostname DEFAULT

If you set the FQDomain to 'yes' in the .conf file and you run it on a computer with the following properties:

- *hostname*: myhost
- IP address: 192.168.1.100
- *domainname*: mycompany.com

An event is created and the three slots are assigned the following values:

"hostname=myhost", "origin=192.168.1.100", "fqhostname=myhost.mycompany.com"

FILENAME

The **FILENAME** keyword indicates the fully qualified file name (including the path) of the log file that contains the message. If you use a single agent to monitor multiple log files, and you need to identify the source of the event, use this keyword to populate an event attribute with the file name. If the message comes from the system log, then mapping is set to EventLog for Windows agents and SysLogD for UNIX logfile agents.

Note: The IBM Tivoli Monitoring path includes an attribute for this keyword.

LABEL

The LABEL keyword indicates the host name of the system where the agent is currently running.

REGEX

The **REGEX** keyword expands to the regular expression that matched the message and caused the event.

Maximum message length

This value is the maximum message length that the Log File agent can receive without truncating the message.

The maximum message length is different for Tivoli Monitoring and Tivoli Netcool/OMNIbus.

Tivoli Monitoring

For events sent to IBM Tivoli Monitoring, the msg attribute is limited to 2048 bytes. Messages greater in length are truncated.

Tivoli Netcool/OMNIbus

For events sent through the Probe for Tivoli EIF to Netcool/OMNIbus, the total size of the event, including the class name and all slots and their values cannot exceed 4096 bytes (extended limit as of version 6.2.2.3 of the agent). For example, in the following sample EIF event, ;END does not count against the 4096-byte limit. However everything else does count against the limit, including the syntactic elements, such as the semicolons, quotation marks, and equal signs.

Class;attr1=value1;attr2=value2;msg='Hello, world';END

Chapter 5. Event filtering and summarization

Use the event filtering and summarization options that you set in the configuration (.conf) file to control how duplicate events are handled by the Tivoli Log File Agent.

When a log is monitored, an event can display multiple times quickly. For example, this repeated logging can occur when the application that produces the log encounters an error and it logs this error continuously until the situation is resolved. When this type of logging occurs, an excessive number of events are sent to the monitoring infrastructure. The volume of events has a negative impact on performance.

Note: The event detection and summarization procedures are supported only on events that are sent to IBM Tivoli Monitoring. You cannot complete these procedures on events that are sent to OMNIbus by FIF

Related reference:

Chapter 3, "Configuration file," on page 23

The Tivoli Log File Agent uses a configuration file that contains configuration options and filters. You create this configuration file and configure the agent instance to use it.

Detecting and filtering duplicate events

You can configure the Tivoli Log File Agent to handle duplicate events.

To alleviate the problem of multiple duplicate events, You define what constitutes a duplicate event by using the DupDetectionKeyAttributes tag in the .conf file. In a comma-separated list, you include one or more defined IBM Tivoli Monitoring attributes that you want to use to determine whether an event is considered a duplicate. In the following example, events with the same message and the same CustomSlot1 are to be considered duplicates:

DupDetectionKeyAttributes=msg,CustomSlot1

Duplicate events are detected from IBM Tivoli Monitoring attributes. Therefore, if you want detection of duplicates to be based on particular slots that you defined, complete the following steps:

- 1. Map the slot value to an IBM Tivoli Monitoring attribute.
- 2. Map that IBM Tivoli Monitoring attribute to the DupDetectionKeyAttributes tag in the .conf file.

Using the following example, where the important slots, eventclass and eventid, are mapped to *CustomSlot1* and *CustomSlot2*:

if you want to specify certain events as duplicate events, in the .conf file, map the IBM Tivoli Monitoring attributes to the DupDetectionKeyAttributes tag as shown here:

DupDetectionKeyAttributes=CustomSlot1,CustomSlot2

Note:

- 1. The CustomSlot attribute names are case-sensitive and so you must enter the names exactly as shown in the preceding example.
- 2. If you do not provide a list of attributes, the values are defaulted to Class and Logname.

The events where these attributes match are considered to be duplicate events by the agent.

Since the duplicate detection is global, it is good practice to pick a set of custom slots to use as keys and use them this way in all format statements. For example, use slots 1-3 for keys. If a format needs only one key but also needs more slots, use slot one to contain the name value and slots four to n to contain the other data.

Summary interval

The duplication detection procedure operates over a time period known as the Summary Interval.

Duplicate events are counted during this interval and then reset when the interval expires. The counter starts the count again beginning at 0 at the start of each new summary interval.

The agent sends a summary event for each event set that it monitors during the interval. The summary event contains the attribute values of the first event that matched. The summary event also contains a count that indicates how many duplicates of that event occurred during the summary interval.

The summary interval is set in the configuration (.conf) file as shown in the following example: EventSummaryInterval=300

The value assigned to the summary interval is in seconds, so in this example, the summary interval is 5 minutes.

Filtering events

If filtering of events is running, the EventFloodThreshold setting in the configuration (.conf) file informs the agent when to send an event.

The following table shows the EventFloodThreshold values:

Table 3. EventFloodThreshold values

EventFloodThreshold values	Description
send_all	The <i>send_all</i> value is the default value. All events are sent even if these events are duplicate events.
send_none	The <i>send_none</i> value means that no individual events are sent. Only the summary events are sent.
send_first	Use the <code>send_first</code> value to send the first event as soon as it is encountered. If duplicates of that first event occur within a specified time, then subsequent duplicates of this first event are not sent. For more information, see "Summary interval."
n integer	Use the <i>n</i> integer value to send only every <i>n</i> occurrence of an event (for example every fifth duplicate) during a specific time. For more information, see "Summary interval."

Summarization attributes

The Event Type and Occurrence Count attributes are used to help summarize events.

When event summarization is enabled, the Event Type and Occurrence Count attributes become meaningful. The Event Type attribute indicates the type of the event, being either an *Event* or a *Summary Event*. General events that correspond to records found in the log on a one-to-one basis are tagged as *Event*. Summary events that are sent at the end of the Summary Interval, are tagged as *Summary Event*.

The Occurrence Count attribute indicates the total amount of duplicate records found in the log for the event. Summary events include this count as it shows the number of events received that matched the summary event during the previous summary interval.

Situations and Summary Events

Regardless of the filter value as described in "Filtering events" on page 44, you always get the summary events at the end of each summary interval, for any event that occurred at least once during that interval. If you are not expecting the summary events your situations can be accidentally triggered. To avoid this accidental triggering of a situation, include a clause in the situation for *Event Type== Event* or *Event Type!= Summary Event*.

Viewing event filtering and summarization in the Tivoli Enterprise Portal

Examples of how data is treated depending on your event filtering and summarization choices.

The agent maintains a cache of the last events received. By default, this cache is 100 in size. If you enable event filtering and summarization for the agent, differences can occur between the number of events in the cache and the number of events sent to IBM Tivoli Monitoring. Additional events in the cache might not reach the designated threshold for sending. Or you might have fewer events in the cache if you selected the **send_all** option. If the **send_all** option is set, an event is sent each time a duplicate occurs. However only one copy of the event is kept in the cache, and the occurrence count is incremented each time the event occurs. To view the events that are sent to IBM Tivoli Monitoring, create a historical view. For information about creating historical views, see Historical Reporting in the *Tivoli Enterprise Portal User's Guide*. You can compare this view with the real-time cache view in the Tivoli Enterprise Portal. You can also use situations to make the same comparison.

The following examples indicate how the same data is treated depending on your choice, if any, of event filtering and summarization. Four agent instances are created, one for each of the four possible values of the EventFloodThreshold setting. The instances have the same name as the value of EventFloodThreshold that they use in the .conf file. All four instances use <code>CustomSlot1</code> as the duplicate detection key, controlled by the following .conf file setting:

DupDetectionKeyAttributes=CustomSlot1

Each instance uses the same format definition, as follows:

```
REGEX DupExample
example: (.*)
example $1 CustomSlot1
msg PRINTF("The example is %s", example)
END
```

All four instances monitor the same text log. The same line of text was added to the monitored log 10 times:

```
example: This is an error
```

In each example, a historical view and the real-time (cache) view is shown. By default, the historical view shows the newest events in the last rows, while the default real-time view of the cache shows the newest events in the first rows. In these examples, the historical view shows the last 1 hour. *CustomSlot2-10* are hidden in these examples.

As new events arrive, you can see them in the cache view. As duplicates of an event arrive, the data is updated in the existing row. When a summary interval elapses, the existing events are converted to summary events and sent. New rows are then added for the next summary interval.

Figure 1 shows the historical view and cache view if you selected the send_all option in the .conf file. The send_all option is the default option. The 10 events and the summary event are shown in the historical view.

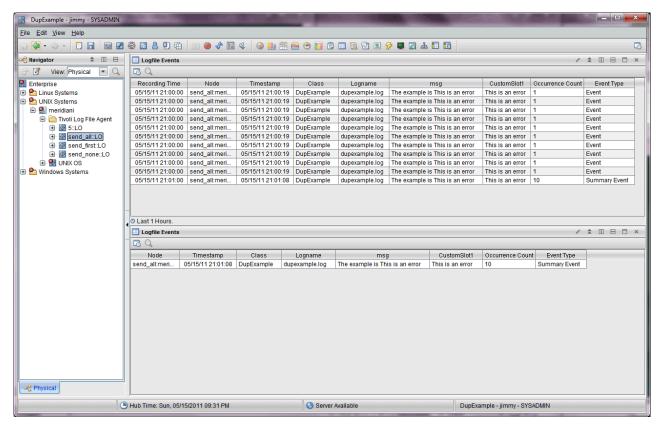


Figure 1. Historical view and cache view when send_all is selected

Figure 2 on page 47 shows the historical view and cache view if you selected the send_none option in the .conf file. Only the summary event is shown in both views.

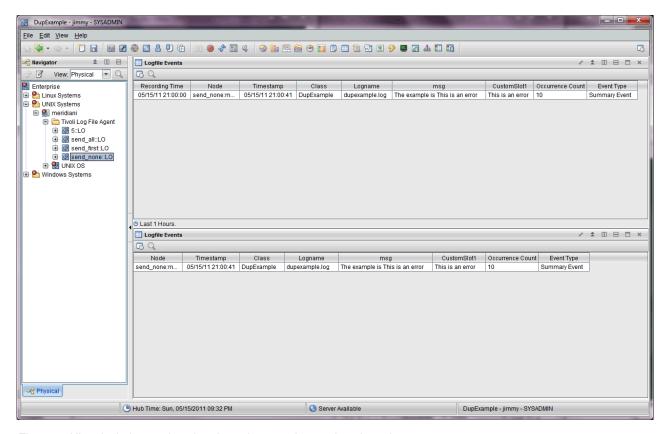


Figure 2. Historical view and cache view when send_none is selected

Figure 3 on page 48 shows the historical view and cache view if you selected the send_first option in the .conf file. The first event received in the summary interval and the summary event for the interval are shown in the historical view.

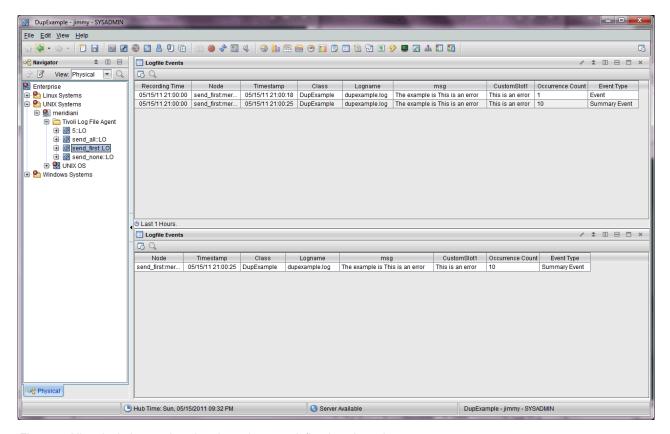


Figure 3. Historical view and cache view when send_first is selected

Figure 4 on page 49 shows the historical view and cache view if you selected the nInteger option in the .conf file and entered a value of 5. An event is shown only when five duplicates of an event (including the first event) are received in the interval. If less than five duplicates are received, no event is shown. If 6, 7, 8, or 9 duplicates are received in the interval, one event is shown. As 10 duplicates were received, 2 events are shown, plus the summary event.

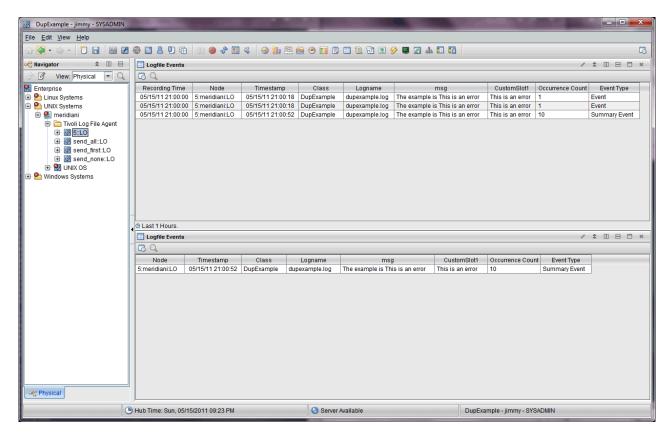


Figure 4. Historical view and cache view when nInteger is selected

Chapter 6. Workspaces reference

A workspace is the working area of the Tivoli Enterprise Portal application window. The Navigator tree contains a list of the workspaces provided by the agent.

About workspaces

Use the Navigator tree to select the workspace you want to see. As part of the application window, the status bar shows the Tivoli Enterprise Portal Server name and port number to which the displayed information applies and the ID of the current user.

When you select an item in the Navigator tree, a default workspace is displayed. When you right-click a Navigator item, a menu that includes a Workspace item is displayed. The Workspace item contains a list of workspaces for that Navigator item. Each workspace has at least one view. Some views have links to other workspaces. You can also use the Workspace Gallery tool as described in the *Tivoli Enterprise Portal User's Guide* to open workspaces.

The workspaces in the Navigator are displayed in a Physical view that shows your enterprise as a physical mapping or a dynamically populated logical view that is agent-specific. You can also create a Logical view. The Physical view is the default view.

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.

The Tivoli Log File Agent provides various default workspaces. These workspaces are displayed in the Navigator tree under the following nodes and subnodes for this monitoring agent:

Log Files

Corresponds to a Log Files instance and contains agent instance-level workspaces.

Tivoli Log File Profile

Tivoli Log File Profile

When multiple instances of the monitoring agent are defined on a system, the top-level node becomes Log Files. The Log Files workspace is undefined at this node. A node for each instance is created called *Instance:Hostname*:LO. A workspace that is called *Instance:Hostname*:LO is associated with the instance node. This workspace is comparable to the Log Files workspace.

Workspace views can be any combination of query-based views, event views, and special purpose views.

Additional information about workspaces

For more information about creating, customizing, and working with workspaces, see "Using workspaces" in the *Tivoli Enterprise Portal User's Guide*.

For a list of the predefined workspaces for this monitoring agent and a description of each workspace, see Predefined workspaces and the information about each individual workspace.

Some attribute groups for this monitoring agent might not be represented in the predefined workspaces or views for this agent. For a full list of the attribute groups, see "Attribute groups for the monitoring agent" on page 57.

Predefined workspaces

The Log File agent provides predefined workspaces, which are organized by Navigator item.

Agent-level navigator items

- · Log Files Navigator item
 - Log Files workspace
 - All Regular Expression statistics workspace
 - Data Collection Status workspace
 - Data Collection Status (remote) workspace
 - High Average CPU Regular Expressions workspace
 - High Total CPU Regular Expressions workspace
 - Logfile Events (remote) workspace
 - Logfile Events (v6) workspace
 - Tivoli Log File Profile (remote) workspace
 - Unmatched Regular Expressions workspace
- · LFAProfiles Navigator item
 - LFAProfiles workspace
- LogfileEvents Navigator item
 - Logfile Events (v6) workspace
 - LogfileEvents workspace

Tivoli Log File Profile (pro) subnode

- Tivoli Log File Profile Navigator item
 - Tivoli Log File Profile workspace
- LogfileProfileEvents Navigator item
 - LogfileProfileEvents workspace

Workspace descriptions

Each workspace description provides information about the workspace such as the purpose and a list of views in the workspace.

Workspaces are listed under Navigator items. When the agent has subnodes, the Navigator items are listed under the subnode.

Log Files Navigator item

The workspace descriptions are organized by the Navigator item to which the workspaces are relevant. **Log Files workspace**

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent.

This workspace contains the following view:

Logfile Events

Shows a table of records that are found in the monitored logs.

All Regular Expression statistics workspace

This workspace provides information about all the regular expressions that are configured in the format file.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in the format file.

Data Collection Status workspace

This workspace provides an overview of data collection of the target systems.

This workspace contains the following views:

Data Collection Status

This view contains information that reflects the status of data collection for all attribute groups that make up this application all at once. Each attribute group is represented by a row in this table. The status for an attribute group reflects the result of the last attempt to collect data for that attribute group. When you check the status information, you can see whether the agent is performing correctly.

Thread Pool Status

This view contains information that reflects the status of an internal thread pool that is used to initialize data collection internally.

Monitored File Status

This view contains information that reflects the status of various files and other data sources that are being monitored by the log agent.

Data Collection Status (remote) workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent. In the (remote) version of this workspace, the RemoteHost attribute indicating the host on which the event originated is also available.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in the format file.

High Average CPU Regular Expressions workspace

This workspace provides information about the regular expressions in the format file that are responsible for the most processor usage in the agent.

This workspace contains the following views:

Top 10 Regular Expressions by average CPU time consumed

This view displays a chart of the 10 regular expressions in the format file that are responsible for the most processor usage in the agent. The X axis is labeled with an index number that indicates the regular expression. This number can be correlated with the Filter Number to determine which actual expression is represented by each index.

Top 10 Regular Expressions by average CPU time consumed

This view displays a table of the 10 regular expressions in the format file that are responsible for the most processor usage in the agent.

High Total CPU Regular Expressions workspace

This workspace provides information about the regular expressions in the format file that are responsible for the most processor usage in the agent.

This workspace contains the following views:

Top 10 Regular Expressions by total CPU time consumed

This view displays a chart of the 10 regular expressions in the format file that are responsible for the most processor usage in the agent. The X axis is labeled with an index number that indicates the regular expression. This number can be correlated with the Filter Number to determine which actual expression is represented by each index.

Top 10 Regular Expressions by total CPU time consumed

This view displays a table of the 10 regular expressions in the format file that are responsible for the most processor usage in the agent.

Logfile Events (remote) workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent. In the (remote) version of this workspace, the RemoteHost attribute indicating the host on which the event originated is also available.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in

Logfile Events (v6) workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent. In the (v6) version of this workspace, the CustomInteger slots are also available.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in the format file.

Tivoli Log File Profile (remote) workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent. In the (remote) version of this workspace, the RemoteHost attribute indicating the host on which the event originated is also available.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in the format file.

Unmatched Regular Expressions workspace

This workspace provides information about the regular expressions in the format file that do not match any patterns in the log. Remove these unmatched expressions if you do not need them. Or optimize the order of the format file so that these unmatched expressions are checked last (formats are searched beginning from the end of the format file)

This workspace contains the following view:

Unmatched Regular Expressions

This view displays a table of the regular expressions in the format file that do not match any patterns in the log. Remove these unmatched expressions if you do not need them. Or optimize the order of the format file so that these unmatched expressions are checked last (formats are searched beginning from the end of the format file).

LFAProfiles Navigator item

The workspace descriptions are organized by the Navigator item to which the workspaces are relevant. **LFAProfiles workspace**

This workspace shows a view of the individual monitoring profiles that are discovered by the agent. For each configuration/format file pair that is discovered in the FILE_DISCOVERY_DIR config directory, a subnode is created.

This workspace contains the following view:

Log File Agent Profiles

This workspace shows a view of the individual monitoring profiles that are discovered by the agent. For each configuration/format file pair that is discovered in the FILE_DISCOVERY_DIR config directory, a subnode is created.

LogfileEvents Navigator item

The workspace descriptions are organized by the Navigator item to which the workspaces are relevant. **Logfile Events (v6) workspace**

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent. In the (v6) version of this workspace, the CustomInteger slots are also available.

This workspace contains the following view:

All Regular Expression statistics

This view provides information about all the regular expressions that are configured in the format file.

LogfileEvents workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent.

This workspace contains the following view:

LogfileEvents

Shows a table of records that are found in the monitored logs.

Tivoli Log File Profile subnode

The predefined workspace descriptions for the subnode are organized by the Navigator item to which the workspaces are relevant.

Tivoli Log File Profile Navigator item

Tivoli Log File Profile workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent.

This workspace contains the following view:

Tivoli Log File Profile

Shows a table of records that are found in the monitored logs.

LogfileProfileEvents Navigator item

LogfileProfileEvents workspace

This workspace shows a view of the agent's cache of events that are found in the monitored logs. Events from all logs are displayed together in the same workspace. If event filtering and summarization are in use, then the presence of a row in this workspace does not necessarily mean that an event was sent. Records must be saved internally for comparison to others to determine when they must be sent. The saved records are visible here even if no events are sent. When the summary interval expires and a summary event is sent, the corresponding row in this workspace changes from Event to Summary Event, rather than adding a row. But the actual event is sent separately. If event filtering and summarization are not in use, then this workspace shows only the events that are sent.

This workspace contains the following view:

Tivoli Log File Profile

Shows a table of records that are found in the monitored logs.

Chapter 7. Attributes reference

Attributes are the application properties that are being measured and reported by the Tivoli Log File Agent.

About attributes

Attributes are organized into attribute groups. Attributes in an attribute group relate to a single object such as an application, or to a single kind of data such as status information.

Attributes in a group can be used in queries, query-based views, situations, policy workflows, take action definitions, and launch application definitions. Chart or table views and situations are two examples of how attributes in a group can be used:

· 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 Properties editor to 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 values you assign to the situation attributes are compared with the values collected by the Log File agent and registers an *event* if the condition is met. You are alerted to events by indicator icons that are displayed in the Navigator.

Additional information about attributes

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

For a list of the attribute groups, a list of the attributes in each attribute group, and descriptions of the attributes for this monitoring agent, see "Attribute groups for the monitoring agent" and "Attributes in each attribute group" on page 58.

Attribute groups for the monitoring agent

The Log File agent contains the following attribute groups. The table name depends on the maximum table name limits of the target database being used for the Tivoli Data Warehouse. If the maximum name is 30 characters, any warehouse table name longer than 30 characters is shortened to 30 characters.

· Attribute group name: LFAProfiles

- Table name: KLOLFAPFLS

Warehouse table name: KLO_LFAPROFILES or KLOLFAPFLS

Attribute group name: Log File RegEx Statistics

- Table name: KLOLOGFRX

Warehouse table name: KLO_LOG_FILE_REGEX_STATISTICS or KLOLOGFRX

• Attribute group name: Log File Status

- Table name: KLOLOGFST

- Warehouse table name: KLO_LOG_FILE_STATUS or KLOLOGFST

Attribute group name: LogfileEvents

- Table name: KLOLOGEVTS

- Warehouse table name: KLO_LOGFILEEVENTS or KLOLOGEVTS
- Attribute group name: LogfileProfileEvents
 - Table name: KLOLOGPEVT
 - Warehouse table name: KLO_LOGFILEPROFILEEVENTS or KLOLOGPEVT
- Attribute group name: Performance Object Status
 - Table name: KLOPOBJST
 - Warehouse table name: KLO_PERFORMANCE_OBJECT_STATUS or KLOPOBJST
- Attribute group name: pro Performance Object Status
 - Table name: KLOPROPOS
 - Warehouse table name: KLO_PRO_PERFORMANCE_OBJECT_STATUS or KLOPROPOS
- Attribute group name: Thread Pool Status
 - Table name: KLOTHPLST
 - Warehouse table name: KLO_THREAD_POOL_STATUS or KLOTHPLST

Attributes in each attribute group

Attributes in each Log File agent attribute group collect data that the agent uses for monitoring.

The descriptions of the attribute groups contain the following information:

Historical group

Whether the attribute group is a historical type that you can roll off to a data warehouse.

Attribute descriptions

Information such as description, type, source, and warehouse name, as applicable, for each attribute in the attribute group.

Some attributes are designated as key attributes. A *key attribute* is an attribute that is used in warehouse aggregation to identify rows of data that represent the same object.

The Source information sometimes uses C programming code syntax for if-then-else clauses to describe how an attribute is derived, for example:

```
(CPU Pct < 0 ) || (Memory Pct < 0 )? 0 : 1
```

This example means that if the CPU_Pct attribute is less than 0 or if the Memory_Pct attribute is less than 0, then the attribute is set to 0. Otherwise, the attribute is set to 1.

LFAProfiles attribute group

Log File Agent Profiles

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the LFAProfiles attribute group: **Node attribute: This attribute is a key attribute.**

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Subnode MSN attribute: This attribute is a key attribute.

Description

The Managed System Name of the subnode agent.

Type

String

Warehouse name

SUBNODE_MSN or SN_MSN

Subnode Affinity attribute

Description

The affinity for the subnode agent.

Type

String

Warehouse name

SUBNODE AFFINITY or SN AFFIN

Subnode Type attribute: This attribute is a key attribute.

Description

The Node Type of this subnode.

Type

String

Warehouse name

SUBNODE_TYPE or SN_TYPE

Subnode Resource Name attribute

Description

The Resource Name of the subnode agent.

Type

String

Warehouse name

SUBNODE_RESOURCE_NAME or SN_RES

Subnode Version attribute

Description

The Version of the subnode agent.

Type

String

Warehouse name

SUBNODE_VERSION or SN_VER

Subnode Config File attribute

Description

The path to the discovered configuration file that caused this subnode to be created.

Type

String

Warehouse name

SUBNODE_CONFIG_FILE or SN_FILE

Subnode Description attribute

Description

User supplied description of this subnode, specified via the SubnodeDescription key in the config file.

Type

String with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• Set the SubnodeDescription keyword in your config file to populate this field (none)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

SUBNODE_DESCRIPTION or SN_DESC

Log File RegEx Statistics attribute group

The Log File RegEx Statistics attribute group contains information that shows the statistics of the log file regular expression search expressions. Regular expressions can be used to filter records or to define records. This attribute group shows information about both types. When the Result Type attribute value is INCLUDE or EXCLUDE, the filter is used to filter records; when the value is BEGIN or END, the filter is used to define records. The CPU measurements are approximations based on the granularity of the data exposed by the operating system. This can result in values of 0.00 when a regular expression takes little time to evaluate. The CPU times should be used to determine the relative cost of regular expressions and to optimize the behavior of specific specific regular expressions.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Log File RegEx Statistics attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Table Name attribute: This attribute is a key attribute.

Description

The name of the log file attribute group.

Type

String

Warehouse name

TABLE_NAME or TBLNAME

Attrib Name attribute: This attribute is a key attribute.

Description

The name of the attribute to which this filter is applied.

Type

String

Warehouse name

ATTRIB NAME or ATRNAME

Filter Number attribute

Description

The sequence number, starting at zero, of the filter being used for the attribute.

Type

Integer (32-bit numeric property)

Warehouse name

FILTER_NUMBER or FLTRNUM

Average Processor Time attribute

Description

The average number of processor seconds used to process the filter for this attribute. The average processor time is the total processor seconds divided by the filter count.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- N/A (-1)
- N/A (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

AVERAGE_PROCESSOR_TIME or CPUTAVG

Total Processor Time attribute

Description

The total number of processor seconds used to process the filter for this attribute. The processor time is cumulative and is truncated, not rounded. This is similar to the Linux /proc/123/task/12/stat where 123 is the PID of the process and 12 is the thread id.

Type

Real number (64-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- N/A (-1)
- N/A (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

TOTAL_PROCESSOR_TIME or CPUTIME

Max Processor Time attribute

Description

The maximum number of processor seconds used for a single filter processing. It is possible that the maximum is zero if the filter has never been used or if each of the filter processing took less than one hundredth of a second.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- N/A (-1)
- N/A (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

MAX_PROCESSOR_TIME or CPUTMAX

Min Processor Time attribute

Description

The minimum number of processor seconds used for a single filter processing. It is possible that the minimum is zero if one of the filter processing took less than one hundredth of a second.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- N/A (-1)
- N/A (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

MIN_PROCESSOR_TIME or CPUTMIN

Filter Count attribute

Description

The number of times the filter has been invoked. Used with the total processor time to compute the average processor time.

Type

Integer (64-bit counter)

Warehouse name

FILTER COUNT or COUNT

Filter Count Matched attribute

Description

The number of times the filter has been invoked and the attribute matched.

Type

Integer (64-bit counter)

Warehouse name

FILTER_COUNT_MATCHED or COUNTMA

Filter Count Unmatched attribute

Description

The number of times the filter has been invoked and the attribute did not match.

Type

Integer (64-bit counter)

Warehouse name

FILTER_COUNT_UNMATCHED or COUNTUN

RegEx Pattern attribute: This attribute is a key attribute.

Description

The regular expression used for the match.

Type

String

Warehouse name

REGEX PATTERN or REGXPAT

Last Matched Time attribute

Description

The last time the filter was used and the result was matched.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli

Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST_MATCHED_TIME or LASTMAT

Last Unmatched Time attribute

Description

The last time the filter was used and the result was unmatched.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• N/A (0000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST_UNMATCHED_TIME or LASTUMA

Result Type attribute

Description

The filter type can be INCLUDE or EXCLUDE to accept or reject the attribute if the filter matches, respectively; the filter type can be BEGIN or END to specify the start or end of a record for multi-line records.

Type

Integer (32-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- UNKNOWN (0)
- INCLUDE (1)
- EXCLUDE (2)
- BEGIN (3)
- END (4)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

RESULT_TYPE or RSTTYPE

Log File Status attribute group

The Log File Status attribute group contains information that reflects the status of log files this agent is monitoring.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Log File Status attribute group: **Node attribute: This attribute is a key attribute.**

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Table Name attribute: This attribute is a key attribute.

Description

The name of the table in which this log is being monitored.

Type

String

Warehouse name

TABLE NAME or TBLNAME

File Name attribute: This attribute is a key attribute.

Description

The name of the monitored file. The file name will be null if the row represents a file pattern to be matched during the next scan.

Type

String

Warehouse name

FILE NAME or FILNAME

RegEx Pattern attribute: This attribute is a key attribute.

Description

The regular expression pattern (if any) that caused this file to be monitored. The regular expression may be the same as the File Name attribute if the regular expression contains no meta characters or if the dot meta character actually matches a dot. If the regular expression is different than the non-null File Name attribute, then the regular expression pattern was used to match the filename.

Type

String

Warehouse name

REGEX_PATTERN or REPATRN

File Type attribute

Description

The type of this file (regular file or pipe).

Type

Integer (32-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- UNKNOWN (0)
- REGULAR FILE (1)
- PIPE (2)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

FILE TYPE or FILTYPE

File Status attribute

Description

The current status of this file.

Type

Integer (32-bit numeric property) with enumerated values. The strings are

displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- N/A (-1)
- OK (0)
- PERMISSION DENIED (1)
- FILE DOES NOT EXIST (2)
- PATH DOES NOT EXIST (3)
- INTERRUPTED SYSTEM CALL (4)
- I/O ERROR (5)
- NO SUCH DEVICE (6)
- BAD FILE NUMBER (9)
- OUT OF MEMORY (12)
- ACCESS DENIED (13)
- RESOURCE BUSY (16)
- NOT A DIRECTORY (20)
- IS A DIRECTORY (21)
- INVALID ARGUMENT (22)
- FILE TABLE OVERFLOW (23)
- TOO MANY OPEN FILES (24)
- TEXT FILE BUSY (26)
- FILE TOO LARGE (27)
- NO SPACE LEFT ON DEVICE (28)
- ILLEGAL SEEK ON PIPE (29)
- READ-ONLY FILE SYSTEM (30)
- TOO MANY LINKS (31)
- BROKEN PIPE (32)
- PASSWORD EXPIRED (50)
- PUBLIC KEY UNVERIFIED (51)
- PUBLIC KEY UNRECOGNIZED (52)
- NO SUPPORTED DESCR (53)
- KEY FILE ACCESS ERROR (126)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

FILE_STATUS or FILSTAT

Num Records Matched attribute

Description

The number of processed records from this log which matched one of the specified patterns.

Type

Integer (32-bit counter)

Warehouse name

NUM_RECORDS_MATCHED or RECMTCH

Num Records Not Matched attribute

Description

The number of processed records from this log which did not match any of the specified patterns.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• UNMATCH LOG NOT ENABLED (-1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

NUM RECORDS NOT MATCHED or RECUNMT

Num Records Processed attribute

Description

The number of records processed from this log since agent start (including ones that are not matches/events).

Type

Integer (32-bit counter)

Warehouse name

NUM_RECORDS_PROCESSED or RECPROC

Current File Position attribute

Description

The current position in bytes into the monitored file. Data up to this has been processed, after it has not. Not applicable to pipes.

Type

Integer (64-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• N/A (-1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

CURRENT FILE POSITION or OFFSET

Current File Size attribute

Description

The current size of the monitored file. Not applicable to pipes.

Type

Integer (64-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• N/A (-1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

CURRENT_FILE_SIZE or FILSIZE

Last Modification Time attribute

Description

The time when the monitored file was last written to. Not applicable to pipes.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• NOT INITIALIZED (0)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST_MODIFICATION_TIME or LASTMOD

Codepage attribute

Description

The language codepage of the monitored file.

Type

String

Warehouse name

CODEPAGE or CODEPG

Remote Host attribute: This attribute is a key attribute.

Description

The remote host name where the monitored file resides. The host name will be N/A if the row represents a local file.

Type

String

Warehouse name

REMOTE_HOST or REMHOST

Related concepts:

"CPU throttling troubleshooting" on page 123

Problems can occur with the CPU throttling of the monitoring agent.

"Format file specifications" on page 33

The format file describes the patterns that the agent looks for to match events in the monitored logs. The format file consists one or more format specifications.

Related tasks:

"Setting RAS trace parameters by using the GUI" on page 105

On Windows systems, you can use the graphical user interface to set trace options.

LogfileEvents attribute group

Events matching configured formats in monitored log files.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the LogfileEvents attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Class attribute: This attribute is a key attribute.

Description

The Class name of the log file event, as defined in the configured format (.fmt) file.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CLASS or TECCLASS

Logname attribute: This attribute is a key attribute.

Description

The name of the log in which the matching record was found.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

LOGNAME

EIFEvent attribute

Description

The contents of the log file event in EIF format.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

EIFEVENT

msg attribute

Description

The contents of the msg slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

MSG

CustomSlot1 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT1 or CUSLOT1

CustomSlot2 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT2 or CUSLOT2

CustomSlot3 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT3 or CUSLOT3

CustomSlot4 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT4 or CUSLOT4

CustomSlot5 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT5 or CUSLOT5

CustomSlot6 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT6 or CUSLOT6

CustomSlot7 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT7 or CUSLOT7

CustomSlot8 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT8 or CUSLOT8

CustomSlot9 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT9 or CUSLOT9

CustomSlot10 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT10 or CUSLOT10

Occurrence Count attribute

Description

The number of times this event occurred over the current flood control summary interval.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (2147483647)
- Value Exceeds Minimum (-2147483648)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is derived: occurrenceCount().

Warehouse name

OCCURRENCE_COUNT or OCOUNT

Event Type attribute

Description

A flag indicating whether the current event is a flood control summary event.

Type

Integer (32-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Event (0)
- Summary Event (1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is derived: isSummaryEvent().

Warehouse name

EVENT TYPE or EVTYPE

CustomInteger1 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER1 or CUINT1

CustomInteger2 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER2 or CUINT2

CustomInteger3 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER3 or CUINT3

RemoteHost attribute

Description

Remote host on which the event originated if using ssh/remote; blank for local system

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

REMOTEHOST or REMHOST

Related reference:

Chapter 3, "Configuration file," on page 23

The Tivoli Log File Agent uses a configuration file that contains configuration options and filters. You create this configuration file and configure the agent instance to use it.

LogfileProfileEvents attribute group

Events matching configured formats in monitored log files.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the LogfileProfileEvents attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

```
Source
```

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Class attribute: This attribute is a key attribute.

Description

The Class name of the log file event, as defined in the configured format (.fmt)

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CLASS or TECCLASS

Logname attribute: This attribute is a key attribute.

Description

The name of the log in which the matching record was found.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

LOGNAME

EIFEvent attribute

Description

The contents of the log file event in EIF format.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

EIFEVENT

msg attribute

Description

The contents of the msg slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

MSG

CustomSlot1 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT1 or CUSLOT1

CustomSlot2 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT2 or CUSLOT2

CustomSlot3 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT3 or CUSLOT3

CustomSlot4 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT4 or CUSLOT4

CustomSlot5 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT5 or **CUSLOT5**

CustomSlot6 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT6 or CUSLOT6

CustomSlot7 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT7 or CUSLOT7

CustomSlot8 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT8 or CUSLOT8

CustomSlot9 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT9 or CUSLOT9

CustomSlot10 attribute

Description

User-defined slot from the EIF event.

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMSLOT10 or CUSLOT10

Occurrence Count attribute

Description

The number of times this event occurred over the current flood control summary interval.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (2147483647)
- Value Exceeds Minimum (-2147483648)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is derived: occurrenceCount().

Warehouse name

OCCURRENCE COUNT or OCOUNT

Event Type attribute

Description

A flag indicating whether the current event is a flood control summary event.

Type

Integer (32-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Event (0)
- Summary Event (1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is derived: isSummaryEvent().

Warehouse name

EVENT TYPE or EVTYPE

CustomInteger1 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER1 or CUINT1

CustomInteger2 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER2 or CUINT2

CustomInteger3 attribute

Description

User-defined slot with integral type from the EIF event.

Type

Integer (64-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- Value Exceeds Maximum (9223372036854775807)
- Value Exceeds Minimum (-9223372036854775808)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Source

The source for this attribute is Log File Data.

Warehouse name

CUSTOMINTEGER3 or CUINT3

RemoteHost attribute

Description

Remote host on which the event originated if using ssh/remote; blank for local system

Type

String

Source

The source for this attribute is Log File Data.

Warehouse name

REMOTEHOST or REMHOST

Performance Object Status attribute group

The Performance Object Status attribute group contains information that reflects the status of other attribute groups so you can see the status of all of the performance objects that make up this application all at once. Each of these other performance attribute groups is represented by a row in this table (or other type of view). The status for an attribute group reflects the result of the last attempt to collect data for that attribute group, which allows you to see whether the agent is performing correctly. Unlike other attribute groups, the Performance Object Status attribute group does not reflect the state of the monitored application. This attribute group is most often used to determine why data is not available for one of the performance attribute groups.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Performance Object Status attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Query Name attribute: This attribute is a key attribute.

Description

The name of the attribute group.

Type

String

Warehouse name

QUERY NAME or ATTRGRP

Object Name attribute

Description

The name of the performance object.

Type

String

Warehouse name

OBJECT NAME or OBJNAME

Object Type attribute

Description

The type of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- WMI (0)
- PERFMON (1)
- WMI ASSOCIATION GROUP (2)
- JMX (3)
- SNMP (4)
- SHELL COMMAND (5)
- JOINED GROUPS (6)
- CIMOM (7)
- CUSTOM (8)
- ROLLUP DATA (9)
- WMI REMOTE DATA (10)
- LOG FILE (11)
- JDBC (12)
- CONFIG DISCOVERY (13)
- NT EVENT LOG (14)
- FILTER (15)
- SNMP EVENT (16)
- PING (17)
- DIRECTOR DATA (18)
- DIRECTOR EVENT (19)
- SSH REMOTE SHELL COMMAND (20)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_TYPE or OBJTYPE

Object Status attribute

Description

The status of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- ACTIVE (0)
- INACTIVE (1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_STATUS or OBJSTTS

Error Code attribute

Description

The error code that is associated with the query.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• NO ERROR (0)

- GENERAL ERROR (1)
- OBJECT NOT FOUND (2)
- COUNTER NOT FOUND (3)
- NAMESPACE ERROR (4)
- OBJECT CURRENTLY UNAVAILABLE (5)
- COM LIBRARY INIT FAILURE (6)
- SECURITY INIT FAILURE (7)
- PROXY SECURITY FAILURE (9)
- NO INSTANCES RETURNED (10)
- ASSOCIATOR QUERY FAILED (11)
- REFERENCE QUERY FAILED (12)
- NO RESPONSE RECEIVED (13)
- CANNOT FIND JOINED QUERY (14)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 1 RESULTS (15)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 2 RESULTS (16)
- QUERY 1 NOT A SINGLETON (17)
- QUERY 2 NOT A SINGLETON (18)
- NO INSTANCES RETURNED IN QUERY 1 (19)
- NO INSTANCES RETURNED IN QUERY 2 (20)
- CANNOT FIND ROLLUP QUERY (21)
- CANNOT FIND ROLLUP ATTRIBUTE (22)
- FILE OFFLINE (23)
- NO HOSTNAME (24)
- MISSING LIBRARY (25)
- ATTRIBUTE COUNT MISMATCH (26)
- ATTRIBUTE NAME MISMATCH (27)
- COMMON DATA PROVIDER NOT STARTED (28)
- CALLBACK REGISTRATION ERROR (29)
- MDL LOAD ERROR (30)
- AUTHENTICATION FAILED (31)
- CANNOT RESOLVE HOST NAME (32)
- SUBNODE UNAVAILABLE (33)
- SUBNODE NOT FOUND IN CONFIG (34)
- ATTRIBUTE ERROR (35)
- CLASSPATH ERROR (36)
- CONNECTION FAILURE (37)
- FILTER SYNTAX ERROR (38)
- FILE NAME MISSING (39)
- SQL QUERY ERROR (40)
- SQL FILTER QUERY ERROR (41)
- SQL DB QUERY ERROR (42)
- SQL DB FILTER QUERY ERROR (43)
- PORT OPEN FAILED (44)
- ACCESS DENIED (45)
- TIMEOUT (46)
- NOT IMPLEMENTED (47)
- REQUESTED A BAD VALUE (48)
- RESPONSE TOO BIG (49)
- GENERAL RESPONSE ERROR (50)
- SCRIPT NONZERO RETURN (51)
- SCRIPT NOT FOUND (52)
- SCRIPT LAUNCH ERROR (53)
- CONF FILE DOES NOT EXIST (54)
- CONF FILE ACCESS DENIED (55)
- INVALID CONF FILE (56)
- EIF INITIALIZATION FAILED (57)

- CANNOT OPEN FORMAT FILE (58)
- FORMAT FILE SYNTAX ERROR (59)
- REMOTE HOST UNAVAILABLE (60)
- EVENT LOG DOES NOT EXIST (61)
- PING FILE DOES NOT EXIST (62)
- NO PING DEVICE FILES (63)
- PING DEVICE LIST FILE MISSING (64)
- SNMP MISSING PASSWORD (65)
- DISABLED (66)
- URLS FILE NOT FOUND (67)
- XML PARSE ERROR (68)
- NOT INITIALIZED (69)
- ICMP SOCKETS FAILED (70)
- DUPLICATE CONF FILE (71)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

ERROR CODE or ERRCODE

Last Collection Start attribute

Description

The most recent time a data collection of this group started.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (00000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST_COLLECTION_START or COLSTRT

Last Collection Finished attribute

Description

The most recent time a data collection of this group finished.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (00000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST COLLECTION FINISHED or COLFINI

Last Collection Duration attribute

Description

The duration of the most recently completed data collection of this group in seconds.

Type

Real number (32-bit counter) with two decimal places of precision

Warehouse name

LAST_COLLECTION_DURATION or COLDURA

Average Collection Duration attribute

Description

The average duration of all data collections of this group in seconds.

Type

Real number (32-bit counter) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

AVERAGE_COLLECTION_DURATION or COLAVGD

Refresh Interval attribute

Description

The interval at which this group is refreshed in seconds.

Type

Integer (32-bit counter)

Warehouse name

REFRESH INTERVAL or REFRINT

Number of Collections attribute

Description

The number of times this group has been collected since agent start.

Type

Integer (32-bit counter)

Warehouse name

NUMBER_OF_COLLECTIONS or NUMCOLL

Cache Hits attribute

Description

The number of times an external data request for this group was satisfied from the cache.

Type

Integer (32-bit counter)

Warehouse name

CACHE HITS or CACHEHT

Cache Misses attribute

Description

The number of times an external data request for this group was not available in the cache.

Type

Integer (32-bit counter)

Warehouse name

CACHE_MISSES or CACHEMS

Cache Hit Percent attribute

Description

The percentage of external data requests for this group that were satisfied from the cache.

Type

Real number (32-bit counter) with two decimal places of precision

Warehouse name

CACHE HIT PERCENT or CACHPCT

Intervals Skipped attribute

Description

The number of times a background data collection for this group was skipped because the previous collection was still running when the next one was due to start.

Type

Integer (32-bit counter)

Warehouse name

INTERVALS_SKIPPED or INTSKIP

pro Performance Object Status attribute group

The Performance Object Status attribute group contains information that reflects the status of other attribute groups so you can see the status of all of the performance objects that make up this application all at once. Each of these other performance attribute groups is represented by a row in this table (or other type of view). The status for an attribute group reflects the result of the last attempt to collect data for that attribute group, which allows you to see whether the agent is performing correctly. Unlike other attribute groups, the Performance Object Status attribute group does not reflect the state of the monitored application. This attribute group is most often used to determine why data is not available for one of the performance attribute groups.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the pro Performance Object Status attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Query Name attribute: This attribute is a key attribute.

Description

The name of the attribute group.

Type

String

Warehouse name

QUERY_NAME or ATTRGRP

Object Name attribute

Description

The name of the performance object.

Type

String

Warehouse name

OBJECT_NAME or OBJNAME

Object Type attribute

Description

The type of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- WMI (0)
- PERFMON (1)
- WMI ASSOCIATION GROUP (2)
- JMX (3)
- SNMP (4)
- SHELL COMMAND (5)
- JOINED GROUPS (6)
- CIMOM (7)
- CUSTOM (8)
- ROLLUP DATA (9)
- WMI REMOTE DATA (10)
- LOG FILE (11)
- JDBC (12)
- CONFIG DISCOVERY (13)
- NT EVENT LOG (14)
- FILTER (15)
- SNMP EVENT (16)
- PING (17)
- DIRECTOR DATA (18)
- DIRECTOR EVENT (19)
- SSH REMOTE SHELL COMMAND (20)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_TYPE or OBJTYPE

Object Status attribute

Description

The status of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- ACTIVE (0)
- INACTIVE (1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_STATUS or OBJSTTS

Error Code attribute

Description

The error code that is associated with the query.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO ERROR (0)
- GENERAL ERROR (1)
- OBJECT NOT FOUND (2)
- COUNTER NOT FOUND (3)
- NAMESPACE ERROR (4)
- OBJECT CURRENTLY UNAVAILABLE (5)

- COM LIBRARY INIT FAILURE (6)
- SECURITY INIT FAILURE (7)
- PROXY SECURITY FAILURE (9)
- NO INSTANCES RETURNED (10)
- ASSOCIATOR QUERY FAILED (11)
- REFERENCE QUERY FAILED (12)
- NO RESPONSE RECEIVED (13)
- CANNOT FIND JOINED QUERY (14)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 1 RESULTS (15)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 2 RESULTS (16)
- QUERY 1 NOT A SINGLETON (17)
- QUERY 2 NOT A SINGLETON (18)
- NO INSTANCES RETURNED IN QUERY 1 (19)
- NO INSTANCES RETURNED IN QUERY 2 (20)
- CANNOT FIND ROLLUP QUERY (21)
- CANNOT FIND ROLLUP ATTRIBUTE (22)
- FILE OFFLINE (23)
- NO HOSTNAME (24)
- MISSING LIBRARY (25)
- ATTRIBUTE COUNT MISMATCH (26)
- ATTRIBUTE NAME MISMATCH (27)
- COMMON DATA PROVIDER NOT STARTED (28)
- CALLBACK REGISTRATION ERROR (29)
- MDL LOAD ERROR (30)
- AUTHENTICATION FAILED (31)
- CANNOT RESOLVE HOST NAME (32)
- SUBNODE UNAVAILABLE (33)
- SUBNODE NOT FOUND IN CONFIG (34)
- ATTRIBUTE ERROR (35)
- CLASSPATH ERROR (36)
- CONNECTION FAILURE (37)
- FILTER SYNTAX ERROR (38)
- FILE NAME MISSING (39)
- SQL QUERY ERROR (40)
- SQL FILTER QUERY ERROR (41)
- SQL DB QUERY ERROR (42)
- SQL DB FILTER QUERY ERROR (43)
- PORT OPEN FAILED (44)
- ACCESS DENIED (45)
- TIMEOUT (46)
- NOT IMPLEMENTED (47)
- REQUESTED A BAD VALUE (48)
- RESPONSE TOO BIG (49)
- GENERAL RESPONSE ERROR (50)
- SCRIPT NONZERO RETURN (51)
- SCRIPT NOT FOUND (52)
- SCRIPT LAUNCH ERROR (53)
- CONF FILE DOES NOT EXIST (54)
- CONF FILE ACCESS DENIED (55)
- INVALID CONF FILE (56)
- EIF INITIALIZATION FAILED (57)
- CANNOT OPEN FORMAT FILE (58)
- FORMAT FILE SYNTAX ERROR (59)
- REMOTE HOST UNAVAILABLE (60)
- EVENT LOG DOES NOT EXIST (61)
- PING FILE DOES NOT EXIST (62)

- NO PING DEVICE FILES (63)
- PING DEVICE LIST FILE MISSING (64)
- SNMP MISSING PASSWORD (65)
- DISABLED (66)
- URLS FILE NOT FOUND (67)
- XML PARSE ERROR (68)
- NOT INITIALIZED (69)
- ICMP SOCKETS FAILED (70)
- DUPLICATE CONF FILE (71)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

ERROR CODE or ERRCODE

Last Collection Start attribute

Description

The most recent time a data collection of this group started.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (00000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST COLLECTION START or COLSTRT

Last Collection Finished attribute

Description

The most recent time a data collection of this group finished.

Type

Timestamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (00000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

LAST_COLLECTION_FINISHED or COLFINI

Last Collection Duration attribute

Description

The duration of the most recently completed data collection of this group in seconds.

Type

Real number (32-bit counter) with two decimal places of precision

Warehouse name

LAST_COLLECTION_DURATION or COLDURA

Average Collection Duration attribute

Description

The average duration of all data collections of this group in seconds.

Type

Real number (32-bit counter) with two decimal places of precision with

enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

• NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

AVERAGE_COLLECTION_DURATION or COLAVGD

Refresh Interval attribute

Description

The interval at which this group is refreshed in seconds.

Type

Integer (32-bit counter)

Warehouse name

REFRESH_INTERVAL or REFRINT

Number of Collections attribute

Description

The number of times this group has been collected since agent start.

Type

Integer (32-bit counter)

Warehouse name

NUMBER OF COLLECTIONS or NUMCOLL

Cache Hits attribute

Description

The number of times an external data request for this group was satisfied from the cache.

Type

Integer (32-bit counter)

Warehouse name

CACHE_HITS or CACHEHT

Cache Misses attribute

Description

The number of times an external data request for this group was not available in the cache.

Type

Integer (32-bit counter)

Warehouse name

CACHE_MISSES or CACHEMS

Cache Hit Percent attribute

Description

The percentage of external data requests for this group that were satisfied from the cache.

Type

Real number (32-bit counter) with two decimal places of precision

Warehouse name

CACHE_HIT_PERCENT or CACHPCT

Intervals Skipped attribute

Description

The number of times a background data collection for this group was skipped because the previous collection was still running when the next one was due to start.

Type

Integer (32-bit counter)

Warehouse name

INTERVALS SKIPPED or INTSKIP

Thread Pool Status attribute group

The Thread Pool Status attribute group contains information that reflects the status of the internal thread pool used to collect data asynchronously.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Thread Pool Status attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Thread Pool Size attribute

Description

The number of threads currently existing in the thread pool.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_SIZE or THPSIZE

Thread Pool Max Size attribute

Description

The maximum number of threads allowed to exist in the thread pool.

Type

Integer (32-bit numeric property) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_MAX_SIZE or TPMAXSZ

Thread Pool Active Threads attribute

Description

The number of threads in the thread pool currently active doing work.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_ACTIVE_THREADS or TPACTTH

Thread Pool Avg Active Threads attribute

Description

The average number of threads in the thread pool simultaneously active doing work.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_AVG_ACTIVE_THREADS or TPAVGAT

Thread Pool Min Active Threads attribute

Description

The smallest number of threads in the thread pool that have simultaneously been active doing work.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_MIN_ACTIVE_THREADS or TPMINAT

Thread Pool Max Active Threads attribute

Description

The peak number of threads in the thread pool that have simultaneously been active doing work.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_MAX_ACTIVE_THREADS or TPMAXAT

Thread Pool Queue Length attribute

Description

The number of jobs currently waiting in the thread pool queue.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_QUEUE_LENGTH or TPQLGTH

Thread Pool Avg Queue Length attribute

Description

The average length of the thread pool queue during this run.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_AVG_QUEUE_LENGTH or TPAVGQL

Thread Pool Min Queue Length attribute

Description

The minimum length the thread pool queue has reached.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD_POOL_MIN_QUEUE_LENGTH or TPMINQL

Thread Pool Max Queue Length attribute

Description

The peak length the thread pool queue has reached.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD POOL MAX QUEUE LENGTH or TPMAXQL

Thread Pool Avg Job Wait attribute

Description

The average time a job spends waiting on the thread pool queue in seconds.

Type

Real number (32-bit gauge) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD POOL AVG JOB WAIT or TPAVJBW

Thread Pool Total Jobs attribute

Description

The number of jobs completed by all threads in the pool since agent start.

Type

Integer (32-bit counter) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO DATA (-1)
- NO DATA (-100)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

THREAD POOL TOTAL JOBS or TPTJOBS

Disk capacity planning for historical data

Disk capacity planning for a monitoring agent is a prediction of the amount of disk space to be consumed by the historical data in each attribute group that is collecting historical data. Required disk storage is an important factor when you are defining data collection rules and your strategy for historical data collection.

The Capacity planning for historical data table provides the following information, which is required to calculate disk space for this monitoring agent:

Table Table name as it is displayed in the warehouse database, if the attribute group is configured to be written to the warehouse. The table name listed here corresponds to the table name in "Attribute groups for the monitoring agent" on page 57.

Attribute group

Name of the attribute group that is used to create the table in the warehouse database if it is short enough to fit in the table naming constraints of the database that is being used for the warehouse. The attribute group name listed here corresponds to the Warehouse table name in "Attribute groups for the monitoring agent" on page 57.

Bytes per row (agent)

Estimate of the record length for each row or instance that is written to the agent disk for historical data collection. This estimate can be used for agent disk space planning purposes.

Database bytes per row (warehouse)

Estimate of the record length for detailed records that are written to the warehouse database, if the attribute group is configured to be written to the warehouse. Detailed records are records that have been uploaded from the agent for long-term historical data collection. This estimate can be used for warehouse disk-space planning purposes.

Aggregate bytes per row (warehouse)

Estimate of the record length for aggregate records that are written to the warehouse database, if the attribute group is configured to be written to the warehouse. Aggregate records are created by the Summarization agent for attribute groups that have been configured for summarization. This estimate can be used for warehouse disk-space planning purposes.

In addition to the information in the tables, you must know the number of rows of data that you plan to collect. An attribute group can have single or multiple rows of data, depending on the application environment that is being monitored. For example, if your attribute group monitors each processor in your computer and you have a dual processor computer, the number of rows is two.

Table 4. Capacity planning for historical data logged by the Log File agent

		Bytes per row	Database bytes per row	Aggregate bytes per row
Table	Attribute group	(agent)	(warehouse)	(warehouse)
KLOLFAPFLS	KLO_LFAPROFILES	2245	2256	2293
KLOLOGEVTS	KLO_LOGFILEEVENTS	6892	6955	7160
KLOLOGPEVT	KLO_LOGFILEPROFILEEVENTS	6892	6955	7160
KLOLOGFRX	KLO_LOG_FILE_REGEX_STATISTICS	798	856	1178
KLOLOGFST	KLO_LOG_FILE_STATUS	688	704	813
KLOPOBJST	KLO_PERFORMANCE_OBJECT_STATUS	352	399	664
KLOPROPOS	KLO_PRO_PERFORMANCE_OBJECT_STATUS	352	399	664
KLOTHPLST	KLO_THREAD_POOL_STATUS	124	168	550

For more information about historical data collection, see Managing historical data in the IBM Tivoli Monitoring Administrator's Guide.

Chapter 8. Situations reference

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 or from the command-line interface using the tacmd commands for situations. You can manage private situations in the private configuration XML file.

About situations

The monitoring agents that you use to monitor your system environment include a set of predefined situations that you can use as-is. You can also 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 Tivoli Log File Agent. You can change the conditions or values being monitored by a predefined situation to the conditions or values best suited to your enterprise.

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

Formula

Formula describing the condition being tested.

Distribution

List of managed systems (operating systems, subsystems, or applications) to which the situation can be distributed. All the Log File agent managed systems are assigned by default.

Expert advice

Comments and instructions to be read in the event workspace.

Action

Command to be sent to the system.

EIF Customize forwarding of the event to an Event Integration Facility receiver. (Available when the Tivoli Enterprise Monitoring Server is configured to forward events.)

Until Options to close the event after a period of time, or when another situation becomes true.

Additional information about situations

The *Tivoli Enterprise Portal 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 and information about each individual situation for this monitoring agent, see "Predefined situations."

Predefined situations

The monitoring agent contains predefined situations.

- Performance Object Status
 - - KLO_Log_Agent_Config_Error
- pro Performance Object Status
 - KLO_Log_Agent_Config_Error

Situation descriptions

Each situation description provides information about the situation that you can use to monitor the condition of systems in your network.

The situation descriptions provide the following information:

Description

Information about the conditions that the situation tests.

Formula

Syntax that contains one or more logical expressions describing the conditions for the situation to monitor.

Distribution

Whether the situation is automatically distributed to instances of the agent or is available for manual distribution.

Run at startup

Whether the situation starts monitoring when the agent starts.

Sampling interval

Number of seconds that elapse between one sample of data that the monitoring agent collects for the server and the next sample.

Situation persistence

Whether the conditions specified in the situation evaluate to "true" for the defined number of occurrences in a row before the situation is raised. The default of one means that no persistence-checking takes place.

Severity

Severity of the predefined events: Warning, Informational, or Critical.

Clearing conditions

Controls when a true situation closes: after a period of time, when another situation is true, or whichever occurs first if both are selected.

KLO_Log_Agent_Config_Error situation

This situation uses data from the Performance Object Status attribute group

KLO_Log_Agent_Config_Error situation

Description

The agent is not configured correctly and is not monitoring anything.

The situation is evaluated for each distinct value of the ATTRGRP attribute.

Formula

*IF *VALUE KLO PERFORMANCE OBJECT STATUS.Object Status *EQ INACTIVE

See "Attributes in each attribute group" on page 58 for descriptions of the attributes in this formula.

Distribution

This situation is automatically distributed to instances of this agent.

Run at startup

Yes

Sampling interval

1 minute

Situation persistence

The number of times the conditions of the situation must occur for the situation to be true is 2.

Error conditions

Warning

Clearing conditions

The situation clears when the condition becomes false.

KLO_PRO_Log_Agent_Config_Error situation

This situation uses data from the pro Performance Object Status attribute group

KLO_PRO_Log_Agent_Config_Error situation

Description

The agent is not configured correctly and is not monitoring anything.

The situation is evaluated for each distinct value of the ATTRGRP attribute.

Formula

*IF *VALUE KLO PRO PERFORMANCE OBJECT STATUS.Object Status *EQ INACTIVE

See "Attributes in each attribute group" on page 58 for descriptions of the attributes in this formula.

Distribution

This situation is automatically distributed to instances of this agent.

Run at startup

Yes

Sampling interval

1 minute 40 seconds

Situation persistence

The number of times the conditions of the situation must occur for the situation to be true is 2.

Error conditions

Warning

Clearing conditions

The situation clears when the condition becomes false.

Chapter 9. Take Action commands reference

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

About Take Action commands

When included in a situation, the command runs 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.

In advanced automation, policies are used to take 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, the Tivoli Enterprise Portal receives return-code feedback, and advanced automation logic responds with subsequent activities that are prescribed by the feedback.

A basic Take Action command shows the return code of the operation in a message box that is displayed after the action is completed or in a log file. After you close this window, no further information is available for this action.

Additional information about Take Action commands

For more information about working with Take Action commands, see *Take Action commands* in the *Tivoli Enterprise Portal User's Guide*.

Predefined Take Action commands

Not all agents have predefined Take Action commands. But you can create Take Action commands for any agent.

The Tivoli Log File Agent does not provide predefined Take Action commands.

Chapter 10. Policies reference

Policies are used as an advanced automation technique for implementing more complex workflow strategies than you can create through simple automation. All agents do not provide predefined policies, but you can create policies for any agent.

A *policy* is a set of automated system processes that can take 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, the Tivoli Enterprise Portal receives return-code feedback, and advanced automation logic responds with subsequent activities prescribed by the feedback.

For more information about working with policies, see *Automation with policies* in the *Tivoli Enterprise Portal 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

Not all agents have predefined policies. But you can create policies for any agent.

The Tivoli Log File Agent does not provide predefined policies.

Chapter 11. Troubleshooting

Problems can be related to IBM Tivoli Monitoring or the specific agent that you are using.

For general troubleshooting information, see the *IBM Tivoli Monitoring Troubleshooting Guide*. For other problem-solving options, see "Support information" on page 130.

You can resolve some problems by ensuring that your system matches the system requirements listed in the Prerequisites topic for the agent in the information center, or in the Requirements topic of the agent user's guide.

The following activities can help you find a solution to the problem you are having:

- "Gathering product information for IBM Software Support"
- "Using logging" on page 100
- "Consulting the lists of identified problems and workarounds" on page 100

Gathering product information for IBM Software Support

Before contacting IBM Software Support about a problem you are experiencing with this product, gather the information shown in Table 5.

Table 5. 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 "Principal trace log files" on page 101 for lists of all trace log files and their locations. For general information about the IBM Tivoli Monitoring environment, see the <i>Tivoli Enterprise Portal User's Guide</i> .
Application log files information	Version number and patch level
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.
	Tivoli Log File Agent
Screen captures	Screen captures of incorrect output, if any
(UNIX systems only) Core dump files	If the system stops on UNIX systems, collect the core dump file from the <code>install_dir/bin</code> directory, where <code>install_dir</code> is the directory where you installed the monitoring agent.

You can use the pdcollect tool to collect the most commonly used information from a system. This tool gathers log files, configuration information, version information, and other data. For more information about using this tool, see the "pdcollect tool" in the *IBM Tivoli Monitoring Troubleshooting Guide*.

For information about working with IBM Software Support, see IBM Support Portal Service Requests and PMRs (http://www.ibm.com/support/entry/portal/Open_service_request/Software/Software_support_(general)).

Using logging

Logging is the primary troubleshooting feature in the Log File agent. *Logging* refers to the text messages and trace data that is generated by the Log File agent. Messages and trace data are sent to a file.

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" for more information.

Consulting the lists of identified problems and workarounds

Known problems are organized into types such as those in the following list to make them easier to locate:

- · Installation and configuration
- General usage and operation
- · Display of monitoring data
- · Take Action commands

Information about symptoms and detailed workarounds for these types of problems is located in "Problems and workarounds" on page 111.

For general troubleshooting information, see the IBM Tivoli Monitoring Troubleshooting Guide.

Trace logging

Trace logs are used to capture information about the operating environment when component software fails to operate as designed.

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 information to learn how to configure and use trace logging:

- "Principal trace log files" on page 101
- "Examples: Using trace logs" on page 104
- "Setting RAS trace parameters by using the GUI" on page 105

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

IBM Software Support personnel use the information captured by trace logging to trace a problem to its source or to determine why an error occurred. All components in the IBM Tivoli Monitoring environment have a default tracing level. The tracing level can be changed on a per-component level to adjust the type of trace information collected, the degree of trace detail, the number of trace logs to be kept, and the amount of disk space used for tracing.

Overview of log file management

Knowing the naming conventions for log files helps you to find the files.

Agent log file naming conventions

Table 6 provides the names, locations, and descriptions of IBM Tivoli Monitoring general RAS1 log files. The log file names for the Log File agent adhere to the following naming convention:

Windows systems

hostname_productcode_instance-name_program_HEXtimestamp-nn.log

Linux and UNIX systems

hostname_productcode_instance-name_program_HEXtimestamp-nn.log

Where:

hostname

Host name of the computer where the monitoring component is running.

productcode

Two-character product code. For Tivoli Log File Agent, the product code is lo.

instance-name

Instance name of the agent.

program

Name of the program being run.

HEXtimestamp

Hexadecimal time stamp representing the time at which the program started.

nn Rolling log suffix.

Principal trace log files

Trace log files are located on various systems.

Table 6 contains locations, file names, and descriptions of trace logs that can help determine the source of problems with agents.

Table 6. Trace log files for troubleshooting agents

System where log is located	File name and path	Description
On the Tivoli Enterprise Monitoring Server	 Windows: The file IBM Tivoli Monitoring timestamp.log in the install_dir\InstallITM path UNIX: The candle_installation.log file in the install_dir/logs path Linux: The 	Provides details about products that are installed. Note: Trace logging is enabled by default. A configuration step is not required to enable this tracing.
candle_installation.log file in the install_dir/logs path		
On the Tivoli Enterprise Monitoring Server	The Warehouse_Configuration.log file is in the following location on Windows systems: install_dir\InstallITM	Provides details about the configuration of data warehousing for historical reporting.

Table 6. Trace log files for troubleshooting agents (continued)

 The name of the RAS log file is as follows: Windows: install_dir\logs\ hostname_ms_timestamp-nn.log UNIX: install_dir/logs/ 	Traces activity on the monitoring server.
hostname_ms_timestamp-nn.log	
• UNIX: install dir/logs/	
hostname_ms_timestamp-nn.log	
• Linux: install_dir/logs/ hostname_ms_timestamp-nn.log	
Note: File names for RAS1 logs include a hexadecimal time stamp.	
Also on UNIX systems, a log with a decimal time stamp is provided: hostname_productcode_timestamp.log and hostname_productcode_timestamp.pid nnnnn in the install_dir/logs path, where nnnnn is the process ID number.	
The name of the RAS log file is as follows:	Traces activity on the portal server.
 Windows: install_dir\logs\ hostname _cq_HEXtimestamp-nn.log UNIX: install_dir /logs/hostname cq HEXtimestamp- 	
nn.log • Linux: install_dir /logs/hostname_cq_HEXtimestamp- nn.log	
Note: File names for RAS1 logs include a hexadecimal time stamp.	
Also on UNIX systems, a log with a decimal time stamp is provided: hostname_productcode_timestamp.log and hostname_productcode_timestamp.pidnnnnn in the install_dir/logs path, where nnnnn is the process ID number.	
The teps_odbc.log file is located in the following path: • Windows: install_dir\InstallITM • UNIX: install_dir/logs	When you enable historical reporting, this log file traces the status of the warehouse proxy agent.
	 Linux: install_dir/logs/hostname_ms_timestamp-nn.log Note: File names for RAS1 logs include a hexadecimal time stamp. Also on UNIX systems, a log with a decimal time stamp is provided: hostname_productcode_timestamp.log and hostname_productcode_timestamp.pid nnnnn in the install_dir/logs path, where nnnnn is the process ID number. The name of the RAS log file is as follows: Windows: install_dir\logs\hostname_cq_HEXtimestamp-nn.log UNIX: install_dir /logs/hostname_cq_HEXtimestamp-nn.log Linux: install_dir /logs/hostname_cq_HEXtimestamp-nn.log Note: File names for RAS1 logs include a hexadecimal time stamp. Also on UNIX systems, a log with a decimal time stamp is provided: hostname_productcode_timestamp.log and hostname_productcode_timestamp.log and hostname_productcode_timestamp.pidnnnnn in the install_dir/logs path, where nnnnn is the process ID number. The teps_odbc.log file is located in the following path: Windows: install_dir\lnstallITM

Table 6. Trace log files for troubleshooting agents (continued)

System where log is located	File name and path	Description
On the computer that hosts the monitoring agent	The RAS1 log files are as follows: • Windows: hostname _lo_instance_name_kloagent_ HEXtimestamp-nn.log in the install_dir\tmaitm6\logs directory • UNIX: hostname_lo_instance_name_ kloagent_ HEXtimestamp-nn.log in the install_dir/logs directory • Linux: hostname_lo_instance_name_ kloagent_ HEXtimestamp-nn.log in the install_dir/logs directory These logs are in the following directories: • Windows: install_dir\tmaitm6\ logs • UNIX: install_dir/logs • Linux: install_dir/logs On Linux systems, the following additional logs are provided: - hostname_lo_timestamp.log - hostname_lo_timestamp.pidnnnn in the install_dir/logs path, where nnnnn is the process ID number	Traces activity of the monitoring agent.
On the computer that hosts the monitoring agent	The agent operations log files are as follows: instance_hostnameL0.LG0 is the current log created when the agent was started. instance_hostname_L0.LG1 is the backup of the previous log. These logs are in the following directory depending on the operating system that you are using: • Windows: install_dir\tmaitm6\ logs • Linux: install_dir/logs • UNIX: install_dir/logs	Shows whether the agent could connect to the monitoring server. Shows which situations are started and stopped, and shows other events while the agent is running. 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 the .LG1 tag to learn the following details regarding the <i>previous</i> monitoring session: • Status of connectivity with the monitoring server • Situations that were running • The success or failure status of Take Action commands

Table 6. Trace log files for troubleshooting agents (continued)

System where log is located	File name and path	Description
On the computer that hosts the monitoring agent	The Take Action command log files are as follows:	Traces activity each time a Take Action command runs. For example,
	• host_lo_instance_ takeactioncommand .log	when a hypothetical start_command Take Action command runs, IBM Tivoli Monitoring generates a
	The logs are in the following directories:	start_command.log file.
	• Windows: install_dir\tmaitm6\ logs	
	• UNIX: install_dir /logs	
	• Linux: install_dir /logs	

Definitions of variables:

- *timestamp* is a time stamp with a format that includes year (y), month (m), day (d), hour (h), and minute (m), as follows: yyyymmdd hhmm
- HEXtimestamp is a hexadecimal representation of the time at which the process was started.
- *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 system, the monitoring agent, or the portal.
- instance refers to the name of the database instance that you are monitoring.
- instance_name refers to the name of the agent instance.
- hostname refers to the name of the computer on which the IBM Tivoli Monitoring component runs.
- *nn* represents the circular sequence in which logs are rotated. this value includes a range from 1 5, by default. The first is always retained because it includes configuration parameters.
- productcode specifies the product code, for example, um for Universal Agent or nt for Windows systems.

For more information about the complete set of trace logs that are maintained on the monitoring server, see the *IBM Tivoli Monitoring Installation and Setup Guide*.

Examples: Using trace logs

You can open trace logs in a text editor to learn some basic facts about your IBM Tivoli Monitoring environment.

IBM Software Support applies specialized knowledge to analyze trace logs to determine the source of problems. The following examples are from the Tivoli Enterprise Monitoring Server log.

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

See the following key points about the preceding excerpts:

- The monitoring server appends the **LO** product code to the server name to form a unique name (SERVER5B:LO) for this instance of the Tivoli Log File Agent. By using this unique name, you can 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 by using the GUI" provide these entries.

On Windows systems, you can use the following alternate method to view trace logs:

- 1. In the Windows Start menu, click Program Files > IBM Tivoli Monitoring > Manage Tivoli Enterprise Monitoring Services. The Manage Tivoli Enterprise Monitoring Services window is displayed.
- 2. Right-click a component and click **Advanced** > **View Trace Log** in the menu. For example, if you want to view the trace log of the Tivoli Log File Agent, right-click the name of that agent in the window. You can also use the viewer to access remote logs.

Note: The viewer converts time stamps in the logs to a format that is easier to read.

RAS trace parameters

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

See "Overview of log file management" on page 100 to ensure that you understand log rolling and can reference the correct log files when you manage log file generation.

Setting RAS trace parameters by using the GUI

On Windows systems, you can use the graphical user interface to set trace options.

About this task

The Tivoli Log File Agent uses RAS1 tracing and generates the logs described in Table 6 on page 101. The default RAS1 trace level is ERROR.

Procedure

- 1. Open the Manage Tivoli Enterprise Monitoring Services window.
- 2. Select Advanced > Edit Trace Parms. The Tivoli Enterprise Monitoring Server Trace Parameters window is displayed.
- 3. Select a new trace setting in the pull-down menu in the Enter RAS1 Filters field or type a valid string.
 - General error tracing. KBB_RAS1=ERROR
 - Intensive error tracing. KBB_RAS1=ERROR (UNIT:logmonitor ALL) (UNIT:kum ALL)

Note: As this example shows, you can set multiple RAS tracing options in a single statement.

- 4. Modify the value for Maximum Log Size Per File (MB) to change the log file size (changes LIMIT value).
- 5. Modify the value for Maximum Number of Log Files Per Session to change the number of log files per startup of a program (changes COUNT value).

- 6. Modify the value for Maximum Number of Log Files Total to change the number of log files for all startups of a program (changes MAXFILES value).
- 7. Optional: Click Y (Yes) in the **KDC_DEBUG Setting** menu to log information that can help you diagnose communications and connectivity problems between the monitoring agent and the monitoring server. The **KDC_DEBUG** setting and the **Maximum error tracing** setting can generate a large amount of trace logging. Use these settings only temporarily, while you are troubleshooting problems. Otherwise, the logs can occupy excessive amounts of hard disk space.
- 8. Click **OK**. You see a message reporting a restart of the monitoring agent so that your changes take effect.

What to do next

Monitor the size of the logs directory. Default behavior can generate a total of 45 - 60 MB for each agent that is running on a computer. For example, each database instance that you monitor can generate 45 - 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 that are pruned automatically, other log types can grow indefinitely, for example, the logs in Table 6 on page 101 that include a process ID number (PID).

Use collector trace logs as an additional source of troubleshooting information.

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

Related concepts:

"CPU throttling troubleshooting" on page 123

Problems can occur with the CPU throttling of the monitoring agent.

Related reference:

"Log File Status attribute group" on page 63

The Log File Status attribute group contains information that reflects the status of log files this agent is monitoring.

Manually setting RAS trace parameters

You can manually edit the RAS1 trace logging parameters.

About this task

The Log File agent uses RAS1 tracing and generates the logs described in Table 6 on page 101. The default RAS1 trace level is ERROR.

Procedure

- 1. Open the trace options file:
 - Windows systems:

install dir\tmaitm6\KLOENV instance name

UNIX systems:

install_dir /config/lo_instance name.config

- 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: KBB_RAS1=ERROR (UNIT:klo ALL) (UNIT:kra ALL)
- 3. Edit the line that begins with KBB_RAS1_LOG= to manage the generation of log files:

- MAXFILES: The total number of files that are to be kept for all startups of a specific program. When this value is exceeded, the oldest log files are discarded. The default value is 9.
- LIMIT: The maximum size, in megabytes (MB) of a RAS1 log file. The 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. The default is
 - PRESERVE: The number of files that are not to be reused in the rolling cycle of one program startup. The 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.

What to do next

Monitor the size of the logs directory. Default behavior can generate a total of 45 - 60 MB for each agent that is running on a computer. For example, each database instance that you monitor can generate 45 - 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 that are pruned automatically, other log types can grow indefinitely, for example, the logs in Table 6 on page 101 that include a process ID number (PID).

Use collector trace logs as an additional source of troubleshooting information.

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

Dynamic modification of trace settings

You can dynamically modify the trace settings for an IBM Tivoli Monitoring component, such as, Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, most monitoring agents, and other components. You can access these components, except for a few monitoring agents, from the tracing utility.

Dynamic modification of the trace settings is the most efficient method, because you can do it without restarting the component. Settings take effect immediately. Modifications by this method are not persistent.

Note: When the component is restarted, the trace settings are read again from the .env file. Dynamically modifying these settings does not change the settings in the .env files. To modify these trace settings permanently, modify them in the .env files.

ras1

Run this command to modify the trace settings for a Tivoli Monitoring component.

The syntax is as follows:

```
ras1 set|list (UNIT|COMP: class name ANY|ALL|Detail|ERROR|Flow|INPUT|Metrics|OUTPUT|STATE)
{(UNIT COMP: class name ANY ALL Detail ERROR Flow INPUT Metrics OUTPUT STATE)}
```

You can specify more than one component class to which to apply the trace settings.

Command options

set

Turns on or off tracing depending upon the value of its parameters. If the parameter is **ANY**, it turns it off. All other parameters turn on tracing based on the specified type or level.

list

Displays the default level and type of tracing that is set by default.

Parameters

The parameters that determine the component classes to which to apply the trace settings are as follows:

COMP: class_name

Modifies the trace setting for the name of the component class, as specified by <code>class_name</code> , for example, <code>COMP:KDH</code>. The output contains trace for the specified class.

UNIT: class name

Modifies the trace setting for any unit that starts with the specified *class_name* value, for example, UNIT: kra. The output contains trace for any unit that begins with the specified filter pattern.

The parameters that determine the trace level and type are as follows:

ALL

Displays all trace levels, including every trace point defined for the component. This setting might result in a large amount of trace, so specify other parameters to exclude unwanted trace. You might require the **ALL** parameter to isolate a problem, which is the equivalent to setting "Error Detail Flow State Input Output Metrics".

ANY

Turns off tracing.

Detail

Displays detailed information about each function.

When entered with the list option, the trace is tagged with Det.

ERROR

Logs internal error conditions.

When entered with the list option, the trace is tagged with ER. The output can also be tagged with EVERYE+EVERYU+ER.

F1 ow

Displays control flow data for each function entry and exit.

When entered with the list option, the trace is tagged with Fl.

INPUT

Displays input data for each function.

When entered with the list option, the trace is tagged with IN.

Metrics

Displays metrics on each function.

When entered with the list option, the trace is tagged with ME.

OUTPUT

Displays output data for each function.

When entered with the list option, the trace is tagged with OUT.

State

Displays the status for each function.

When entered with the list option, the trace is tagged with St.

Example

If you enter ras1 set (COMP:KDH ALL) (COMP:ACF1 ALL) (COMP:KDE ALL), the trace utility turns on all levels of tracing for all the files and functions for which KDH, ACF1, and KDE are the classes.

```
kbbcrel.c, 400, May 29 2007, 12:54:43, 1.1, *
kbbcrn1.c, 400, May 29 2007, 12:54:42, 1.1, * kdhb1de.c, 400, May 29 2007, 12:59:34, 1.1, KDH kdh0med.c, 400, May 29 2007, 12:59:24, 1.1, KDH
kdhsrej.c, 400, May 29 2007, 13:00:06, 1.5, KDH
kdhb1fh.c, 400, May 29 2007, 12:59:33, 1.1, KDH
kdhbloe.c, 400, May 29 2007, 12:59:38, 1.2, KDH
kdhs1ns.c, 400, May 29 2007, 13:00:08, 1.3, KDH
kbbacdl.c, 400, May 29 2007, 12:54:27, 1.2, ACF1
kbbaclc.c, 400, May 29 2007, 12:54:27, 1.4, ACF1
kbbacli.c, 400, May 29 2007, 12:54:28, 1.11, ACF1
vkdhsfcn.c, 400, May 29 2007, 13:00:11, 1.1, KDH kdhserq.c, 400, May 29 2007, 12:59:53, 1.1, KDH
kdhb1pr.c, 400, May 29 2007, 12:59:39, 1.1, KDH
kdhsgnh.c, 400, May 29 2007, 12:59:49, 1.1, KDH
kdh0uts.c, 400, May 29 2007, 12:59:23, 1.1, KDH
kdhsrsp.c, 400, May 29 2007, 13:00:13, 1.2, KDH
kdhs1rp.c, 400, May 29 2007, 13:00:12, 1.1, KDH
kdhscsv.c, 400, May 29 2007, 12:59:58, 1.9, KDH
kdebbac.c, 400, May 29 2007, 12:56:50, 1.10, KDE
```

Turning on tracing

To use the tracing utility, you must use a local logon credential for the computer. This tracing method uses the IBM Tivoli Monitoring Service Console. Access the Service Console by using a web browser.

About this task

When you start the Service Console, information is displayed about the components that are currently running on that computer. For example, these components are listed as follows:

- Tivoli Enterprise Portal Server: cnp
- Monitoring Agent for Windows OS: nt
- Tivoli Enterprise Monitoring Server: ms

After you log on, you can type a question mark (?) to display a list of the supported commands. Use the **ras1** command to modify trace settings. If you type this command in the field provided in the Service Console window and click **Submit**, the help for this command is displayed.

Procedure

1. Open a web browser and enter the URL to access the Service Console.

```
http://hostname:1920
```

where *hostname* is the IP address or host name of the computer on which the IBM Tivoli Monitoring component is running.

2. Click the hyperlink associated with the component for which you want to modify its trace settings.

Note: In the previous view, if you want to modify tracing for the Tivoli Enterprise Monitoring Server, select **IBM Tivoli Monitoring Service Console** under **Service Point:** system.your host name_ms.

- 3. Enter a user ID and password to access the system. This ID is any valid user that has access to the system.
- 4. Enter the command to turn on the required level of trace for the specified component classes or units.

```
ras1 set (UNIT COMP: class_name ALL Flow ERROR Detail INPUT Metrics OUTPUT STATE)
{(UNIT|COMP: class name ALT|Flow|ERROR|Detail|INPUT|Metrics|OUTPUT|STATE)}
```

For example, to turn on the control flow trace for the KDE, the command is: ras1 (COMP:KDE Flow)

Turning off tracing

You can use the IBM Tivoli Monitoring Service Console to run the ras1 command and dynamically turn off tracing.

Procedure

1. Open a web browser and enter the URL to access the Service Console.

```
http://hostname:1920
```

where *hostname* is the IP address or host name of the computer on which the IBM Tivoli Monitoring component is running.

- 2. Click the hyperlink associated with the component for which you want to modify its trace settings.
- 3. Enter a user ID and password to access the system. This ID is any valid user that has access to the system.
- 4. Enter the command to turn off the required level of trace for the specified component classes or units.

```
ras1 set (UNIT COMP: class name ANY)
{(UNIT | COMP: class_name ANY)}
```

For example, to turn off tracing for the kbbcrcd class of the Windows OS agent, the command is: ras1 set (UNIT:kbbcrcd ANY)

Setting trace parameters for the Tivoli Enterprise Console server

In addition to the trace information captured by IBM Tivoli Monitoring, you can also collect additional trace information for the Tivoli Enterprise Console components that gather event server metrics.

About this task

To collect this information, modify the .tec_diag_config file on the Tivoli Enterprise Console event server. Use the steps in the following procedure to modify the event server trace parameters.

Procedure

- 1. Open the \$BINDIR/TME/TEC/.tec diag config file in an ASCII editor.
- 2. Locate the entries that configure trace logging for the agent components on the event server. Two entries are included, one for tec reception and one for tec rule:

```
# to debug Agent Utils
tec reception Agent Utils error /tmp/tec reception
# to debug Agent Utils
tec rule Agent Utils error /tmp/tec rule
```

3. To gather additional trace information, modify these entries to specify a trace level of trace2:

```
# to debug Agent Utils
tec reception Agent Utils
                           trace2 /tmp/tec reception
# to debug Agent Utils
tec rule Agent Utils
                          trace2 /tmp/tec rule
```

4. In addition, modify the Highest_level entries for tec_rule and tec_reception:

```
tec reception Highest level trace2
tec rule Highest level trace2
```

Problems and workarounds

The known problems and workarounds are organized into types of problems that might occur with the Log File agent, for example installation and configuration problems and workspace problems.

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

For general troubleshooting information, see the IBM Tivoli Monitoring Troubleshooting Guide.

Installation and configuration troubleshooting

Problems can occur during installation, configuration, and uninstallation of the agent.

The problems and solutions in Table 7 can occur during installation, configuration, and uninstallation of the agent.

Table 7. Problems and solutions for installation and configuration

Problem	Solution
(UNIX only) During a command-line installation, you choose to install a component that is currently installed, and you see the following warning: WARNING - you are about to install the SAME version of "component_name" where component_name is the name of the component that you are attempting to install. Note: This problem affects UNIX command-line installations. If you monitor only Windows environments, you see this problem if you choose to install a product component (for example, a monitoring server) on a UNIX system.	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 currently installed.
Diagnosing problems with product browse settings (Windows systems only).	When you have problems with browse settings, complete the following steps:
	1. Click Start > Programs > IBM Tivoli Monitoring > Manage Tivoli Enterprise Monitoring Services. The Manage Tivoli Enterprise Monitoring Services window is displayed.
	2. Right-click the Windows agent and select Browse Settings . A text window is displayed.
	3. Click Save As and save the information in the text file.
	If requested, you can forward this file to IBM Software Support for analysis.
A message similar to "Unable to find running CMS on CT_CMSLIST" in the log file is displayed.	If a message similar to "Unable to find running CMS on CT_CMSLIST" is displayed in the log file, the agent cannot connect to the monitoring server. Confirm the following points:
	Do multiple network interface cards (NICs) exist on the system?
	If multiple NICs exist on the system, find out which one is configured for the monitoring server. Ensure that you specify the correct host name and port settings for communication in the IBM Tivoli Monitoring environment.

Table 7. Problems and solutions for installation and configuration (continued)

Problem

The system is experiencing high CPU usage.

High CPU while using the agent may be caused be one or more of these issues:

- Large amounts of data being written to the log files being monitored, which requires a lot of processing and can cause event storms.
- Poor ordering of regular expressions in the format file. The format file is processed from the beginning to the end. If a record has to be compared with all or almost regular expression in order to find a match CPU usage is affected.
- Using regular expressions that take a long time to compute.

Tivoli Log File Agent installation fails on 64-bit WINDOWS Systems

Note:

This issue applies only to users who want to install the Tivoli Log File Agent onto a system that has:

- A 64-bit WINDOWS operating system
- A previous version of the IBM Tivoli Monitoring 32-bit/64-bit compatibility package
- · An IBM Tivoli Monitoring agent (other than the Tivoli Log File Agent) installed

The IBM Tivoli Monitoring installer prevents the installation of the Tivoli Log File Agent onto a 64-bit WINDOWS system that has a previous version of the IBM Tivoli Monitoring 32-bit/64-bit compatibility package. The IBM Tivoli Monitoring installer cannot upgrade the IBM Tivoli Monitoring 32-bit/64-bit compatibility package.

Solution

To help alleviate the issue of high CPU usage while using the agent:

- Follow best practices for writing efficient regular expressions
- Place expressions that are likely to match the most records as far towards the end of the format file as possible.
- Place *DISCARD* statements at the end of the format file to discard unwanted records.
- If sending EIF events, ensure that the EIF cache is large enough to store the event storm.
- Where possible, minimize the use of multi-line records.

Agent process: View the memory usage of the KLOCMA process. If CPU usage seems to be excessive, restart the monitoring agent.

Network cards: The network card configurations can decrease the performance of a system. Each stream of packets that a network card receives (assuming that it is a broadcast or destined for the under-performing system) must generate a CPU interrupt and transfer the data through the I/O bus. If the network card in question is a bus-mastering card, work can be offloaded and a data transfer between memory and the network card can continue without using CPU processing power. Bus-mastering cards are 32-bit and are based on PCI or EISA bus architectures.

Install the IBM Tivoli Monitoring 32-bit/64-bit compatibility package (component AC) from the base IBM Tivoli Monitoring 6.2.2.2 DVD using one of the following methods:

- Run setup.exe and respond to the prompts.
- Use the IBM Tivoli Monitoring remote deployment feature.

If you use the IBM Tivoli Monitoring remote deployment feature, you:

- Add the IBM Tivoli Monitoring 6.2.2.2 version of the AC component to the deployment depot on the Tivoli Enterprise Monitoring Server.
- Deploy the Tivoli Monitoring 32-bit/64-bit compatibility package (component AC) from the deployment depot on the Tivoli Enterprise Monitoring Server.

For instructions about performing installations using setup.exe and the IBM Tivoli Monitoring remote deployment feature, see the IBM Tivoli Monitoring Installation and Setup Guide.

Table 7. Problems and solutions for installation and configuration (continued)

Problem	Solution
The configuration panel is blank on 64-bit Windows systems where the Tivoli Enterprise Monitoring Agent Framework (component GL) is version 06.23.00.00 or 06.23.01.00.	Check the GL component version by running kincinfo -t GL from a Windows command line. Example: %CANDLE_HOME%\InstallITM\kincinfo -t GL If the GL component version is 06.23.00.00 or 06.23.01.00, take one of the following actions:
	• Preferred action: Upgrade the Windows OS Agent to Version 6.2.3 Fix Pack 2.
	• Alternate action: Install the Agent Compatibility (AC) component from the IBM Tivoli Monitoring V6.2.3 Fix Pack 1 media. See Installing the Agent Compatibility (AC) component (http://pic.dhe.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.3fp1/itm623FP1_install199.htm#acpinstall).

Table 8. General problems and solutions for uninstallation

Problem	Solution
On Windows systems, uninstallation of IBM Tivoli Monitoring fails to uninstall the entire environment.	Be sure that you follow the general uninstallation process described in the <i>IBM Tivoli Monitoring Installation and Setup Guide</i> :
	Remove Tivoli Enterprise Monitoring Server Application support by completing the following steps:
	a. Use Manage Tivoli Enterprise Monitoring Services.
	b. Select Tivoli Enterprise Monitoring Server.
	c. Right-click and select Advanced.
	d. Select Remove TEMS application support.
	e . Select the agent to remove its application support.
	2. Uninstall the 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.
	3. Uninstall IBM Tivoli Monitoring.
The way to remove inactive managed systems (systems whose status is OFFLINE) from the Navigator tree in the	Use the following steps to remove, but not uninstall, an offline managed system from the Navigator tree:
portal is not obvious.	1. Click the Enterprise icon in the Navigator tree.
	2. Right-click, and then click Workspace > Managed System Status .
	3. Right-click the offline managed system, and select Clear offline entry.
	To uninstall the monitoring agent, use the procedure described in the <i>IBM Tivoli Monitoring Installation and Setup Guide</i> .

Table 8. General problems and solutions for uninstallation (continued)

Table 6. dericial problems and solutions for uninstaliation	,
Problem	Solution
IBM Tivoli Monitoring might not be able to generate a unique name for monitoring components because of the truncation of names that the product automatically generates.	If the agent supports multiple instances, IBM Tivoli Monitoring automatically creates a name for each monitoring component by concatenating the subsystem name, host name, and product code separated by colons (subsystem_name:hostname:KLO). Note: When you monitor a multinode system, such as a database, IBM Tivoli Monitoring adds a subsystem name to the concatenated name, typically a database instance name.
	The length of the name that IBM Tivoli Monitoring generates is limited to 32 characters. Truncation can result in multiple components having the same 32-character name. If this problem happens, shorten the <i>hostname</i> portion of the name as follows:
	1. Open the configuration file for the monitoring agent, which is located in the following path:
	• On Windows: install_dir\tmaitm6\ Kproduct_codeCMA.INI. For example, the product code for the Monitoring Agent for Windows OS is NT. The file name is KNTCMA.INI.
	• On UNIX and Linux: itm_home/config/ 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 that 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 KLO, 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.
The software inventory tag for the agent on UNIX and Linux systems is not removed during uninstallation of the agent.	After uninstalling the agent, manually remove the file named <i>full name of agent</i> .cmptag from the \$CANDLEHOME/properties/version/ directory.

Table 8. General problems and solutions for uninstallation (continued)

Problem	Solution
When the agent is installed using group deployment, deploygroup was run multiple times. The group deployment starts and completes successfully, but there were multiple entries in the Deploy Status Summary workspace on the Tivoli Enterprise Portal. When the command tried to install multiple times, the additional installations were queued and then were in failed state though the agent was deployed successfully. Note:	There is no solution at this time.
• When the bundle group contains a single bundle and the deployment group contains more than one member (managed system of the same type as AIX or Linux), the deployment is successful on both systems.	
When the bundle group contains more than one bundle and the deploy group contains single or multiple members, the deployment will be executed on each group member (managed system) depending on the members present in the bundle group and deploy group.	
• The command creates a transaction for each XX bundle for each target system; the bundle matching the operating system for the deployment member is processed successfully; and remaining transactions were in a queued or failed state.	

Remote deployment troubleshooting

Problems can occur with remote deployment and removal of agent software using the Agent Remote Deploy process.

Table 9 contains problems and solutions related to remote deployment.

Table 9. Remote deployment problems and solutions

Problem	Solution
While you are using the remote deployment feature to install the Tivoli Log File Agent, an empty command window is displayed on the target computer. This problem occurs when the target of remote deployment is a Windows computer. (For more information about the remote deployment feature, see the <i>IBM Tivoli Monitoring Installation and Setup Guide.</i>)	Do not close or modify this window. It is part of the installation process and is 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 occur when you attempt the remote removal process immediately after you restart 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.

Agent troubleshooting

A problem can occur with the agent after it has been installed.

Table 10 on page 116 contains problems and solutions that can occur with the agent after it is installed.

Table 10. Agent problems and solutions

Problem	Solution
Log data accumulates too rapidly.	Check the RAS trace option settings, which are described in "Setting RAS trace parameters by using the GUI" on page 105. The trace option settings that you can set on the KBB_RAS1= and KDC_DEBUG= lines potentially generate large amounts of data.
When using the itmcmd agent commands to start or stop this monitoring agent, you receive the following error message:	Include the command option -0 to specify the instance to start or stop. The instance name must match the name used for configuring the agent. For example:
MKCIIN0201E Specified product is not configured.	./itmcmd agent -o Test1 start lo
	For more information about using the itmcmd commands, see the <i>IBM Tivoli Monitoring Command Reference</i> .

Problem

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 are displayed in the portal.

Solution

IBM Tivoli Monitoring products use Remote Procedure Call (RPC) to define and control product behavior. RPC is the mechanism that a client process uses 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 protocol (or delivery mechanism) for RPCs that you want.

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 by 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 another type of monitoring server process. 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 than 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) if each Tivoli Enterprise Monitoring Server on that image reports to a different hub. By definition, one Tivoli Enterprise Monitoring Server hub is available per monitoring enterprise, so this architecture limit has been reduced to one Tivoli Enterprise Monitoring Server per system image.

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

Continued on next row.

Table 10. Agent problems and solutions (continued)

Problem	Solution
Continued from previous row.	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 the EPHEMERAL IP.PIPE process. (This process is IP.PIPE configured with the 'EPHEMERAL:Y' keyword in the KDC_FAMILIES / KDE_TRANSPORT environment variable). The number of ephemeral IP.PIPE connections per system image has no limitation. 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.)
I cannot find my queries.	Agents that include subnodes display their queries within the element in the Query Editor list that represents the location of the attribute group. The queries are most often found under the name of the subnode, not the name of the agent.

Table 10. Agent problems and solutions (continued)

Problem	Solution
No events are visible in the Tivoli Enterprise Portal even	This problem can be caused be one of these issues:
though the agent is started.	The agent is not configured to send events to IBM Tivoli Monitoring:
	Ensure that the <i>Send ITM Events</i> configuration value is set to "yes" in the command or checked in the GUI.
	The configuration file cannot be found. The performance object status attribute group reports CONF FILE DOES NOT EXIST:
	Ensure the value for the configuration file configuration setting is correct. It must be a fully qualified path to the configuration file and the agent must have sufficient access rights to read the file.
	The format file cannot be found. The performance object status attribute group reports CANNOT OPEN FORMAT FILE:
	Ensure the value for the format file configuration setting is correct. It must be a fully qualified path to the format file and the agent must have sufficient access rights to read the file.
	• There is a parsing error in the format file. The performance object status attribute group reports FORMAT FILE SYNTAX ERROR. The agent's error log identifies the line of the format file where the parsing error occurred:
	Correct the syntax error in the format file.
	No patterns in the format file match records read from the log source:
	Configure the agent instance to use an unmatch file and examine the contents of the unmatch log. Use a regular expression test tool to help write expressions that match the records written to the unmatch log. Update the format file.

Table 10. Agent problems and solutions (continued)

Problem	Solution
The KLO_Log_Agent_Config_Error situation is true for the Tivoli Log File Agent instance.	The KLO_Log_Agent_Config_Error situation is true if any of the following issues occur:
	The agent is not configured to send events to IBM Tivoli Monitoring:
	Ensure that the <i>Send ITM Events</i> configuration value is set to "yes" in the command or checked in the GUI.
	The configuration file cannot be found. The performance object status attribute group reports CONF FILE DOES NOT EXIST:
	Ensure the value for the configuration file configuration setting is correct. It must be a fully qualified path to the configuration file and the agent must have sufficient access rights to read the file.
	The format file cannot be found. The performance object status attribute group reports CANNOT OPEN FORMAT FILE:
	Ensure the value for the format file configuration setting is correct. It must be a fully qualified path to the format file and the agent must have sufficient access rights to read the file.
	There is a parsing error in the format file. The performance object status attribute group reports FORMAT FILE SYNTAX ERROR. The agent's error log identifies the line of the format file where the parsing error occurred:
	Correct the syntax error in the format file.
	No patterns in the format file match records read from the log source:
	Configure the agent instance to use an unmatch file and examine the contents of the unmatch log. Use a regular expression test tool to help write expressions that match the records written to the unmatch log. Update the format file.

Table 10. Agent problems and solutions (continued)

Problem	Solution
Some events are initially visible in the Tivoli Enterprise Portal, but no new events are generated even though the log sources contain records that match expressions in the format file.	If the agent is configured to send EIF events, but no EIF receiver is configured or running (such as an OMNIbus probe) to receive the events, the agent's EIF reception log fills and no events can be processed.
	If you do not want EIF events sent, complete the following steps:
	1. Stop the agent.
	2. Reconfigure the agent and set the <i>Send EIF Events to OMNIbus</i> configuration value to "No".
	3. Start the agent.
	If you want EIF events sent, complete the following steps:
	1. Ensure that the EIF receiver is running.
	2. Verify the EIF settings in the agent's configuration file.
	3. If too many events are sent, the EIF event cache may not be large enough to handle the event storm. Increase the value of the BufEvtMaxSize configuration value in the agent's configuration file.
syslogd does not support named pipes on some HP-UX operating systems	There is no workaround.
The Automatically initialize UNIX syslog configuration option uses syslogd. Because some HP-UX operating systems do not support named pipes, the Automatically initialize UNIX syslog feature does not work on these HP-UX operating systems.	
An agent configured to send EIF events, stops processing events at a time when there are a large number of events generated and appears to hang.	The EIF event cache may not be large enough to handle the event storm. Increase the value of the BufEvtMaxSize configuration value in the agent's configuration file.
Log record does not match the expected regular expression. The event generated does not have the expected event class or event attributes.	Expressions are processed from the end of the format file to the beginning. The first expression that matches a record is the one that is used. Ensure that the more generic expressions are at the beginning of the file and the more specific ones at the end.
While using LogSources or RegexLogSources, the file is not being monitored.	Setting the KBB_RAS1=ERROR (UNIT:kumpdcm STATE) creates a trace entry every time a file is matched against the LogSources pattern and FileComparisonMethod. Example output: Directory /logs/data/ File notsmoke.log did not match pattern smoke.*\.log File <smoke03.log> skipped, not more recently updated than previous file <smoke01.log></smoke01.log></smoke03.log>
When using RegexLogSources, if more than one subdirectory has meta characters how can you tell what regular expression and filename is being used.	The Log File Status attribute group shows information about each log file being monitored. For each file being monitored, the regular expression pattern used to select the file is shown. If the log file pattern with the RegexLogSources was specified incorrectly, the regular expression pattern shown in the Log File Status attribute group is different from the one specified in the configuration (.conf) file and a warning trace entry is be written.

Table 10. Agent problems and solutions (continued)

Problem	Solution
On Windows systems, when the LogSources and RegexLogSources statement in the .conf file specify files in the root directory without the drive letter, the files to be monitored are not found. For example: LogSources=\temp* RegexLogSources=/.*/reg4ex/.*.log	Include the drive letter of the file path in the LogSources and RegexLogSources statement. For example: LogSources=C:\temp* and RegexLogSources=C:/.*/ reg4ex/.*.log. If a drive letter is not specified for the root directory of LogSources or RegexLogSources, the agent assumes that the drive letter is C: even if the system is configured to have a different default drive letter.
If the EIF configuration is set to connection oriented, ConnectionMode=co or ConnectionMode=connection_oriented in the Log File agent configuration file, and the EIF receiver is down when the agent starts, the remote receiver does not respond for approximately two minutes. The agent is not monitoring the logs during this time. If Profiles/subnodes are in use, and each .conf file is configured the same way, these delays are sequential.	To avoid this issue, ensure the EIF receiver is up and running when the agent starts, or use the connection_less option for event delivery.
On HP/UX 11.11, some systems default to only allowing 65 threads per process. The Log File Agent uses a thread for each file that it is monitoring, as well as a number of other threads performing background housekeeping tasks. When using subnodes or monitoring a large number of files, it is possible to exceed this operating system-imposed limit, which causes the monitoring of some files to fail.	To avoid this issue, use the HP/UX sam tool to configure the kernel parameter max_thread_proc to a larger value. The recommended value is 500 to allow enough threads for a large configuration.
Not all log files that are specified are being monitored.	 Check the Log File Status attribute group. It shows information about each log file that is being monitored. In particular: Verify that the file is one of the log files that is being monitored. If it is being monitored, verify that the File Status attribute is OK (0). If it is OK, verify that the Current File Position is less than the Current File Size as the log file agent might have already read the entire file. Verify that for one of the files that is being monitored, the RegEx Pattern attribute matches the file that is specified in the configuration (.conf) file that represents the missing file. If all log files are not in the Log File Status attribute group, check the trace log files for a limit on system resources. For example, each log file that is being monitored is a separate thread and a UNIX system might have a ulimit -u (max user processes) which is too small. ***** File server process startup failed for file address file name ***** MultiThreaded File server process startup failed for file file name ***** File server process startup failed for file name
There may be issues with Czech language support when viewing help files on some Firefox browser versions.	Use Firefox browser version 18 or later.

CPU throttling troubleshooting

Problems can occur with the CPU throttling of the monitoring agent.

Table 11. CPU throttling problems and solutions

Problem	Solution
While using CPU throttling, a review of the log file does not indicate if the CPU usage limits on the monitoring agent are limiting events.	The Log File Status attribute group shows information about each log file that is being monitored. In particular, for each file that is being monitored, the difference between the Current® File Position and the Current File Size show the amount of data the log file monitor has yet to process. This amount of data that remains to be processed, gives an indication of how far behind the log file agent is. Setting the KBB RAS1=ERROR (UNIT: kumpcpu STATE)
	will create a trace entry every time a thread is actively slowed. The trace entry shows the number of process seconds used, the elapse time, and the utilization. The wait time in milliseconds is the number of milliseconds needed to delay meeting the CPU throttling goal. The number of hot threads are the number of active threads currently doing CPU-intensive work.
	Example output: Spent 0.589910000 sec processor, 5.890441194 sec elapsed, 5.0073%; Waiting 8 millisec; 1 hot threads; From (kumpiafi.c,1161"KUMP_ConstructAttrFilterInfo")
While using CPU throttling, you cannot see the CPU usage for the agent.	Setting the KBB_RAS1=ERROR (UNIT: kumpcpu METRICS STATE) creates a trace entry every minute showing the processor utilization of each process. All platforms can determine the processor utilization of a process. For example, the Windows Task Manager or the Linux top command can be used to determine the process information.
	Example output: Process processor utilization 3.6311% (8.71 / 120.00); Process interval utilization 3.6311% (8.71 / 120.00); 2 cores
While using CPU throttling, you cannot see if the CPU usage limits for the agent are exceeded.	An error trace is issued when the CPU throttling maximum is exceeded for an interval. If the processor utilization is within 10% of the goal, it is considered an informational trace message. If the processor utilization exceeds the goal by 10%, it is considered an error trace message.
	Example output: Informational: Interval processor utilization 11.1372% (26.72 / 120.00) exceeds processor utilization goal 10 with 2 cores

Related tasks:

"Setting RAS trace parameters by using the GUI" on page 105

On Windows systems, you can use the graphical user interface to set trace options.

Related reference:

"Log File Status attribute group" on page 63

The Log File Status attribute group contains information that reflects the status of log files this agent is monitoring.

Remote log file monitor troubleshooting

Problems can occur with agent communications to the remote system for remote log file monitoring.

Table 12. Remote log file monitor problems and solutions

Problem	Solution
No events or data is read from the remote log file.	Check the Log File Status File Status. If it shows OK, then the file is being monitored.
The Log file Status does not show OK File Status.	Check the File Status that relate to remote files:
	PASSWORD EXPIRED expired password.
	PUBLIC KEY UNRECOGNIZED A certificate cannot be read (bad passphrase or bad file).
	PUBLIC KEY UNVERIFIED A certificate cannot be verified.
	NO SUPPORTED DESCR A remote system does not support any of the authentication methods that are described in "Supported SSH capabilities and features" on page 13.
	SOCKET_TIMEOUT A communications time out. Possibly from a firewall or a broken connection.
	KEY FILE ACCESS ERROR A public or private key file cannot be found on the local system.
	HOSTNAME UNRESOLVED The domain name server (DNS) cannot resolve the remote host name.
	FAMILY UNSUPPORTED The IP address of the host name that is returned from DNS is not an IP supported family (IPv4 versus IPv6).
Log File Status does not show one of the values in the previous two troubleshooting entries for File Status.	From the system that is running the Log File agent, use SSH, SFTP, or SCP with the remote parameters (userid, password, certification, filename) and verify that the file can be read.
None of the previous troubleshooting entries help resolve the issue.	Turn on the following trace: ERROR (UNIT: KUMPLIBSSH STATE). This setting produces a trace of all libssh2 communication between the Log File agent and the remote system. Look up any nonzero return codes for the APIs. For more information about using trace logs, see "Trace logging" on page 100.

Workspace troubleshooting

Problems can occur with general workspaces and agent-specific workspaces.

Table 13 contains problems and solutions related to workspaces.

Table 13. Workspace problems and solutions

Problem	Solution
The process application components are available, but the Availability status shows PROCESS_DATA_NOT_AVAILABLE.	This problem occurs because the PerfProc performance object is disabled. When this condition exists, IBM Tivoli Monitoring cannot collect performance data for this process. Use the following steps to confirm that this problem exists and to resolve it:
	1. In the Windows Start menu, click Run .
	2. Type perfmon.exe in the Open field of the Run window. The Performance window is displayed.
	3. Click the plus sign (+) in the toolbar. The Add Counters window is displayed.
	4. Look for Process in the Performance object menu.
	5. Complete one of the following actions:
	If you see Process in the menu, the PerfProc performance object is enabled and the problem is coming from a different source. You might need to contact IBM Software Support.
	If you do not see Process in the menu, use the Microsoft utility from the Microsoft.com Operations website to enable the PerfProc performance object.
	The Process performance object becomes visible in the Performance object menu of the Add Counters windows, and IBM Tivoli Monitoring is able to detect Availability data.
	6. Restart the monitoring agent.
The name of the attribute does not display in a bar chart or graph view.	When a chart or graph view that includes the attribute is scaled to a small size, a blank space is displayed instead of a truncated name. To see the name of the attribute, expand the view of the chart until sufficient space is available to display all characters of the attribute name.
At the end of each view, you see the following Historical workspace KFWITM220E error: Request failed during execution.	Ensure that you configure all groups that supply data to the view. In the Historical Configuration view, ensure that data collection is started for all groups that supply data to the view.

Table 13. Workspace problems and solutions (continued)

Problem	Solution
You start collection of historical data but the data cannot be seen.	 Use the following managing options for historical data collection: Basic historical data collection populates the Warehouse with raw data. This type of data collection is turned off by default. For information about managing this feature including how to set the interval at which data is collected, see Managing historical data in the IBM Tivoli Monitoring Administrator's Guide. By setting a more frequent interval for data collection, you reduce the load on the system incurred every time data is uploaded. Use the Summarization and Pruning agent to collect specific amounts and types of historical data. Historical data is not displayed until the Summarization and Pruning monitoring agent begins collecting the data. By default, this agent begins collection at 2 a.m. daily. At that point, data is visible in the workspace view. For information about how to modify the default collection settings, see Managing historical data in the IBM Tivoli Monitoring Administrator's Guide.
Historical data collection is unavailable because of incorrect queries in the Tivoli Enterprise Portal.	The Sort By, Group By, and First/Last functions column are not compatible with the historical data collection feature. Use of these advanced functions makes a query ineligible for historical data collection. Even if data collection has started, you cannot use the time span feature if the query for the chart or table includes column functions or advanced query options (Sort By, Group By, First / Last). To ensure support of historical data collection, do not use the Sort By, Group By, or First/Last functions in your queries. For information about the historical data collection function, See <i>Managing historical data</i> in the <i>IBM Tivoli Monitoring Administrator's Guide</i> or the Tivoli Enterprise Portal online help.
When you use a long process name in the situation, the process name is truncated.	Truncation of process or service names for situations in the Availability table in the portal display is the expected behavior. The maximum name length is 100 bytes.
Regular (non-historical) monitoring data fails to be displayed.	Check the formation of the queries you use to gather data. For example, look for invalid SQL statements.
Navigator items and workspace titles are labeled with internal names such as Kxx:KXX0000 instead of the correct names (such as Disk), where XX and xx represent the two-character agent code.	Ensure that application support has been added on the monitoring server, portal server, and portal client. For more information about installing application support, see <i>Installing and enabling application support</i> in the <i>IBM Tivoli Monitoring Installation and Setup Guide</i> .

Situation troubleshooting

Problems can occur with situations and situation configuration.

Table 14 contains problems and solutions for situations.

Table 14. Situation problems and solutions

Problem	Solution
Monitoring activity requires too much disk space.	Check the RAS trace logging settings that are described in "Setting RAS trace parameters by using the GUI" on page 105. For example, trace logs grow rapidly when you apply the ALL logging option.
Monitoring activity requires too many system resources.	"Disk capacity planning for historical data" on page 89 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 appears to be incorrect. For example, if you were monitoring a Linux system, the 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. In the example, you can select the % Memory Free attribute and avoid the need for math operators.
You want to change the appearance of situations when they are displayed in the navigation tree.	 Right-click an item in the navigation tree. Click Situations in the menu. The Situation Editor window is displayed. Select the situation that you want to modify. Use the State menu to set the status and appearance of the Situation when it triggers. Note: The State setting is not related to severity settings in the Tivoli Enterprise Console.
When a situation is triggered in the Event Log attribute group, it remains in the Situation Event Console as long as the event ID entry is present in the Event Log workspace. When this event ID entry is removed from the Event Log workspace on the Tivoli Enterprise Portal, the situation is also cleared even if the actual problem that caused the event is not resolved, and the event ID entry is also present in the Windows Event Viewer.	A timeout occurs on the cache of events for the NT Event Log group. Increase the cache time of Event Log collection to meet your requirements by adding the following variable and timeout value to the KpcENV file for the agent (where pc is the two-letter product code): CDP_NT_EVENT_LOG_CACHE_TIMEOUT=3600 This variable determines how long events from the NT Event Log are kept.
An event has not been received by your event consumer, or has been received and is truncated.	Adjust the EventMaxSize attribute in the agent configuration file.
The value of a CustomInteger attribute is not as expected	Check that the value being mapped to the CustomInteger attribute is an integer. For example mapping the following values to a CustomInteger attribute will give unexpected results: • Mapping the string "twenty" to a CustomInteger
	 attribute will result in a zero value Mapping the floating point number "1.9" to a CustomInteger attribute will result in a value of 1 Mapping the string "2 thousand" to a CustomInteger attribute will result in a value of 2

Table 14. Situation problems and solutions (continued)

Problem	Solution
For a situation that uses the 'MISSING' operator and is distributed to a remote agentless monitoring subnode, no indication is displayed in the Tivoli Enterprise Portal or in the Situation Event Console when the situation becomes true.	The MISSING predicate is currently not supported on subnodes. If a situation with a MISSING predicate is distributed to a subnode, the agent cannot tell which subnode or node the event is occurring on. It inserts the system name as the origin node for the event and returns. When the event reaches the Tivoli Enterprise Portal Server, the origin node does not match the system name of the subnode where the situation is associated, so the event is dropped.
If the Expert Advice for a situation contains a hyperlink to an external website (for example, a Microsoft TechNet website) and you click the hyperlink, the website opens in an external window. However, the external window stops responding.	The external window responds after you close the Preview window and Situation Editor window.
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 not displayed, confirm that the monitoring server has been seeded for the agent. If not, seed the server, as described in the <i>IBM Tivoli Monitoring Installation and Setup Guide</i> .
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 required.
The situation did not activate at startup.	Manually recycle the situation as follows:
	1. Right-click the situation and select Stop Situation .
	2. Right-click the situation and select Start Situation .
	Note: You can permanently avoid this problem by selecting the Run at Startup check box 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 did not occur even though the predicate was correctly specified.	Check the logs, reports, and workspaces.
A situation fires on an unexpected managed object.	Confirm that you distributed and started the situation on the correct managed system.
The product did not distribute the situation to a managed system.	Click the Distribution tab and check the distribution settings for the situation.

Table 14. Situation problems and solutions (continued)

Problem	Solution
The situation does not fire.	This problem can be caused when 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 Formula area. The Show formula window is displayed.
	a. Confirm the following details in the Formula area 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 numeric values in the formula match your monitoring goal.
	b. (Optional) Select the Show detailed formula check box 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.
	 (Optional) In the Formula area of the Formula tab, temporarily assign numeric values that 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 numeric values to valid levels so that you do not generate excessive monitoring data based on your temporary settings.
	For additional information about situations that do not fire, see <i>Situations are not firing</i> in the <i>IBM Tivoli Monitoring Troubleshooting Guide</i> .
Situation events are not displayed in the Events Console view of the workspace.	Associate the situation with a Navigator item. Note: The situation does not need to be displayed in the workspace. It is sufficient that the situation is associated with any Navigator item.
You do not have access to a situation.	Note: You must have administrator privileges to complete these steps.
	1. Click 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 role. 4. Click OK.

Table 14. Situation problems and solutions (continued)

Problem	Solution
A managed system seems to be offline.	 Select Physical View and click the Enterprise Level of the navigator tree. Click 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 the status of the specific system or application.

Take Action commands troubleshooting

Problems can occur with Take Action commands.

Table 15 contains problems and solutions that can occur with Take Action commands.

When each Take Action command runs, it generates a log file listed in Table 6 on page 101.

Table 15. Take Action commands problems and solutions

Problem	Solution
Take Action commands often require several minutes to complete.	Allow several minutes. If you do not see a message advising you of completion, try to run the command manually.
Situations fail to trigger Take Action commands.	Attempt to manually run the Take Action command in the Tivoli Enterprise Portal. If the Take Action command works, look for configuration problems in the situation. See "Situation troubleshooting" on page 127. If the Take Action command fails, for general information about troubleshooting Take Action commands, see the <i>IBM Tivoli Monitoring Troubleshooting Guide</i> .

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

The following websites contain troubleshooting information:

- Go to the IBM Software Support website (http://www.ibm.com/support/entry/portal/software) and follow the instructions.
- Go to the Application Performance Management Wiki (http://www.ibm.com/developerworks/servicemanagement/apm/index.html). Feel free to contribute to this wiki.

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 the IBM Support Assistant website (http://www.ibm.com/software/support/isa).

Appendix A. Event mapping

The Tivoli Event Integration Facility (EIF) interface is used to forward situation events to Tivoli Netcool/OMNIbus or Tivoli Enterprise Console.

EIF events specify an event class, and the event data is specified as name-value pairs that identify the name of an event slot and the value for the slot. An event class can have subclasses. IBM Tivoli Monitoring provides the base event class definitions and a set of base slots that are included in all monitoring events. Agents extend the base event classes to define subclasses that include agent-specific slots. For Log File agent events, the event classes correspond to the agent attribute groups, and the agent-specific slots correspond to the attributes in the attribute group.

The situation editor in the Tivoli Enterprise Portal can be used to perform custom mapping of data to EIF slots instead of using the default mapping described in this topic. For more information about EIF slot customization, see the *Tivoli Enterprise Portal User's Guide*.

Tivoli Enterprise Console requires that event classes and their slots are defined in BAROC (Basic Recorder of Objects in C) files. Each agent provides a BAROC file that contains event class definitions for the agent and is installed on the Tivoli Enterprise Monitoring Server in the TECLIB directory (install_dir/cms/TECLIB for Windows systems and install_dir/tables/TEMS_hostname/TECLIB for UNIX systems) when application support for the agent is installed. The BAROC file for the agent and the base BAROC files provided with Tivoli Monitoring must also be installed onto the Tivoli Enterprise Console. For details, see "Setting up event forwarding to Tivoli Enterprise Console" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Each of the event classes is a child of KLO_Base and is defined in the klo.baroc (version 06.30.00) file. The KLO_Base event class can be used for generic rules processing for any event from the Tivoli Log File Agent.

For events that are generated by situations in the LFAProfiles attribute group, events are sent by using the ITM_KLO_LFAPROFILES event class. This event class contains the following slots:

node: STRING

timestamp: STRINGsubnode_msn: STRINGsubnode_affinity: STRINGsubnode_type: STRING

• subnode_resource_name: STRING

subnode_version: STRING subnode_config_file: STRING subnode_description: STRING

subnode_description_enum: STRING

For events that are generated by situations in the Log File RegEx Statistics attribute group, events are sent by using the ITM_KLO_LOG_FILE_REGEX_STATISTICS event class. This event class contains the following slots:

· node: STRING

timestamp: STRINGtable_name: STRINGattrib_name: STRING

• filter_number: INTEGER

average_processor_time: REAL

· average_processor_time_enum: STRING

• total_processor_time: REAL

• total_processor_time_enum: STRING

max_processor_time: REAL

• max_processor_time_enum: STRING

• min_processor_time: REAL

• min_processor_time_enum: STRING

• filter_count: REAL

filter_count_matched: REALfilter_count_unmatched: REAL

• regex_pattern: STRING

• last_matched_time: STRING

last_matched_time_enum: STRING

last_unmatched_time: STRING

last_unmatched_time_enum: STRING

• result_type: INTEGER

• result_type_enum: STRING

For events that are generated by situations in the Log File Status attribute group, events are sent by using the ITM_KLO_LOG_FILE_STATUS event class. This event class contains the following slots:

· node: STRING

• timestamp: STRING

table_name: STRING

• file_name: STRING

regex_pattern: STRING

file_type: INTEGER

file_type_enum: STRING

file_status: INTEGER

file_status_enum: STRING

• num_records_matched: INTEGER

num_records_not_matched: INTEGER

num_records_not_matched_enum: STRING

num_records_processed: INTEGER

current_file_position: REAL

current_file_position_enum: STRING

current_file_size: REAL

• current_file_size_enum: STRING

· last_modification_time: STRING

· last_modification_time_enum: STRING

codepage: STRING

For events that are generated by situations in the LogfileEvents attribute group, events are sent by using the ITM_KLO_LOGFILEEVENTS event class. This event class contains the following slots:

· node: STRING

• timestamp: STRING

• klo_class: STRING

logname: STRING

• eifevent: STRING

klo_msg: STRING

· customslot1: STRING

customslot2: STRING

customslot3: STRING

customslot4: STRING

customslot5: STRING

customslot6: STRING

customslot7: STRING

customslot8: STRING

· customslot9: STRING

· customslot10: STRING

occurrence_count: INTEGER

· occurrence_count_enum: STRING

event_type: INTEGER

event_type_enum: STRING

• custominteger1: REAL

custominteger1_enum: STRING

custominteger2: REAL

custominteger2_enum: STRING

custominteger3: REAL

· custominteger3_enum: STRING

· remotehost: STRING

For events that are generated by situations in the LogfileProfileEvents attribute group, events are sent by using the ITM_KLO_LOGFILEPROFILEEVENTS event class. This event class contains the following slots:

· node: STRING

timestamp: STRING

klo_class: STRING

logname: STRING

• eifevent: STRING

klo_msg: STRING

• customslot1: STRING

customslot2: STRING

customslot3: STRING

customslot4: STRING

customslot5: STRING

· customslot6: STRING

customslot7: STRING

customslot8: STRING

• customslot9: STRING

customslot10: STRING

• occurrence_count: INTEGER

· occurrence_count_enum: STRING

event_type: INTEGER

• event_type_enum: STRING

custominteger1: REAL

custominteger1_enum: STRING

• custominteger2: REAL

• custominteger2_enum: STRING

custominteger3: REAL

custominteger3_enum: STRING

· remotehost: STRING

For events that are generated by situations in the Performance Object Status attribute group, events are sent by using the ITM_KLO_PERFORMANCE_OBJECT_STATUS event class. This event class contains the following slots:

node: STRING

· timestamp: STRING

• query_name: STRING

object_name: STRING

object_type: INTEGER

object_type_enum: STRING

· object_status: INTEGER

• object_status_enum: STRING

error_code: INTEGER

• error code enum: STRING

• last_collection_start: STRING

· last_collection_start_enum: STRING

last_collection_finished: STRING

last_collection_finished_enum: STRING

• last_collection_duration: REAL

average_collection_duration: REAL

· average_collection_duration_enum: STRING

· refresh_interval: INTEGER

number_of_collections: INTEGER

• cache_hits: INTEGER

cache_misses: INTEGER

cache_hit_percent: REAL

intervals_skipped: INTEGER

For events that are generated by situations in the pro Performance Object Status attribute group, events are sent by using the ITM_KLO_PRO_PERFORMANCE_OBJECT_STATUS event class. This event class contains the following slots:

node: STRING

timestamp: STRINGquery_name: STRINGobject_name: STRING

• object_type: INTEGER

• object_type_enum: STRING

· object_status: INTEGER

• object_status_enum: STRING

error_code: INTEGER

• error_code_enum: STRING

• last_collection_start: STRING

· last_collection_start_enum: STRING

last_collection_finished: STRING

• last_collection_finished_enum: STRING

• last_collection_duration: REAL

average_collection_duration: REAL

· average_collection_duration_enum: STRING

• refresh_interval: INTEGER

• number_of_collections: INTEGER

cache_hits: INTEGERcache_misses: INTEGERcache_hit_percent: REALintervals_skipped: INTEGER

For events that are generated by situations in the Thread Pool Status attribute group, events are sent by using the ITM_KLO_THREAD_POOL_STATUS event class. This event class contains the following slots:

· node: STRING

timestamp: STRING

thread_pool_size: INTEGER

thread_pool_size_enum: STRING

• thread_pool_max_size: INTEGER

thread_pool_max_size_enum: STRING

thread_pool_active_threads: INTEGER

• thread_pool_active_threads_enum: STRING

thread_pool_avg_active_threads: REAL

thread_pool_avg_active_threads_enum: STRING

thread_pool_min_active_threads: INTEGER

thread_pool_min_active_threads_enum: STRING

thread_pool_max_active_threads: INTEGER

• thread_pool_max_active_threads_enum: STRING

thread_pool_queue_length: INTEGER

thread_pool_queue_length_enum: STRING

thread_pool_avg_queue_length: REAL

• thread_pool_avg_queue_length_enum: STRING

• thread_pool_min_queue_length: INTEGER

thread_pool_min_queue_length_enum: STRING

thread_pool_max_queue_length: INTEGER

• thread_pool_max_queue_length_enum: STRING

thread_pool_avg_job_wait: REAL

- thread_pool_avg_job_wait_enum: STRING
- thread_pool_total_jobs: INTEGER
- thread_pool_total_jobs_enum: STRING

Appendix B. Windows 2008 event log

This section describes how the Tivoli Log File Agent monitors events from the Windows event log

The Tivoli Log File Agent continues to use the WINEVENTLOGS configuration (.conf) file option to monitor events from the Windows event log. The agent monitors a comma-separated list of event logs as shown in the following example:

WINEVENTLOGS=System, Security, Application

The Tivoli Log File Agent also continues to use the WINEVENTLOGS=All setting. The All setting refers to the following standard event logs: Security, Application, System, Directory, Domain Name System (DNS), and File Replication Service (FRS) that come with Windows versions before 2008. However, all the event logs on the system are not checked.

There is a configuration file tag called UseNewEventLogAPI. This tag allows the event log (Windows event log 2008 or later) to access any of the new logs added by Microsoft, and any Windows event logs created by other applications or created by the user. The new logs are listed by the WINEVENTLOGS keyword.

In the following example, the UseNewEventLogAPI tag is set to y.

UseNewEventLogAPI=y WINEVENTLOGS=Microsoft-Windows-Hyper-V-Worker-Admin

In this example, the Microsoft-Windows-Hyper-V/Admin is monitored on a Windows system that has the Hyper-V role.

In the Windows event log, each event has the following fields in this order:

- The date in the following format: month, day, time, and year.
- The event category is an integer.
- · Event Level
- Windows security ID. Any spaces in the Windows security ID are replaced by an underscore if SpaceReplacement=TRUE in the configuration (.conf) file.

Note: SpaceReplacement=TRUE is the default if you set UseNewEventLogAPI to y in the (.conf) file (designated that you are using the 2008 event log).

- Windows source. Any spaces in the Windows source are replaced by an underscore if SpaceReplacement=TRUE in the configuration (.conf) file.
- Windows event log keywords. Any spaces in the Windows event log keywords are replaced by an underscore if SpaceReplacement=TRUE in the configuration (.conf) file.

Note: The keyword field described here is new to the Windows 2008 version of event log. It did not exist in the previous event log, and so its presence prevents you from reusing your old Event Log format statements directly. They need to be modified to account for this additional field.

- · Windows event identifier
- · Message text

For example, when an administrative user logs on to a Windows 2008 system, an event is generated in the Security log indicating the privileges that are assigned to the new user session:

```
Mar 22 13:58:35 2011 1 Information N/A Microsoft-Windows-
Security-Auditing Audit_Success 4672 Special privileges assigned to new logon.
S-1-5-21-586564200-1406810015-1408784414-500 Account Name:
Administrator Account Domain: MOLDOVA Logon ID:
```

0xc39cb8e Privileges: SeSecurityPrivilege
SeBackupPrivilege SeRestorePrivilege
SeTakeOwnershipPrivilege SeDebugPrivilege
SeSystemEnvironmentPrivilege SeLoadDriverPrivilege
SeImpersonatePrivilege

To capture all the events that been created by the Microsoft-Windows-Security-Auditing event source, you write a format statement as shown here:

For the previous example event, the following example indicates the values that are assigned to slots:

```
timestamp=Mar 22 13:58:35 2011
severity=Information
login=N/A
eventsource=Microsoft-Windows-Security-Auditing
eventid=4672
msg="Special privileges assigned to new logon.
S-1-5-21-586564200-1406810015-1408784414-500 Account Name:
Administrator Account Domain: MOLDOVA Logon ID:
0xc39cb8e Privileges: SeSecurityPrivilege
SeBackupPrivilege SeRestorePrivilege
SeTakeOwnershipPrivilege SeDebugPrivilege
SeSystemEnvironmentPrivilege SeLoadDriverPrivilege
SeImpersonatePrivilege
```

Since it is difficult to anticipate exactly what these events look like, a useful approach to writing your regular expressions is to capture the actual events in a file. Then you can examine the file, choose the events you would like the agent to capture, and write regular expressions to match these events. To capture all the events from your Windows event log, use the following steps:

1. Create a format file that contains only one pattern that does not match anything, as shown in the following example:

```
REGEX NoMatch
This doesn't match anything
```

2. Add the following setting to the configuration (.conf) file:

```
UnmatchLog=C:/temp/evlog.unmatch
```

3. Run the agent and capture some sample events.

Appendix C. Documentation library

A variety of publications are relevant to the use of the Tivoli Log File Agent.

The *IBM Tivoli Monitoring*, *OMEGAMON XE*, and *Composite Application Manager products: Documentation Guide* contains information about accessing and using publications. You can find the Documentation Guide in the following information centers:

- IBM Tivoli Monitoring and OMEGAMON® XE (http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp)
- IBM Tivoli Composite Application Manager (http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/index.jsp)

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 in the information center** 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.

Tivoli Log File Agent library

The documentation for this agent and other product components is located in the IBM Tivoli Monitoring Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp).

One document is specific to the Log File agent. The Tivoli Log File Agent User's Guide provides agent-specific information for configuring, using, and troubleshooting the Log File agent.

The Offering Guide also provides information about installing and configuring the component products in the offering.

The **Prerequisites** topic in the information center contains information about the prerequisites for each component.

Use the information in the user's guide for the agent with the *Tivoli Enterprise Portal User's Guide* to monitor Application log files resources.

Prerequisite publications

To use the information in this publication effectively, you must have some prerequisite knowledge.

See the following publications to gain the required prerequisite knowledge:

- IBM Tivoli Monitoring Administrator's Guide
- IBM Tivoli Monitoring Agent Builder User's Guide
- IBM Tivoli Monitoring Command Reference
- IBM Tivoli Management Services on z/OS: Configuring the Tivoli Enterprise Monitoring Server on z/OS
- IBM Tivoli Monitoring Installation and Setup Guide
- IBM Tivoli Monitoring High Availability Guide for Distributed Systems
- IBM Tivoli Monitoring: Messages
- IBM Tivoli Monitoring Troubleshooting Guide

- IBM Tivoli Monitoring Universal Agent User's Guide
- IBM Tivoli Universal Agent API and Command Programming Reference Guide
- IBM Tivoli Monitoring: i5/OS™ Agent User's Guide
- IBM Tivoli Monitoring: Linux OS Agent User's Guide
- IBM Tivoli Monitoring: UNIX OS Agent User's Guide
- IBM Tivoli Monitoring: UNIX Logs OS Agent User's
- IBM Tivoli Monitoring: Windows OS Agent User's Guide
- Tivoli Enterprise Portal User's Guide
- IBM Tivoli Performance Analyzer User's Guide
- IBM Tivoli Warehouse Proxy Agent User's Guide
- IBM Tivoli Warehouse Summarization and Pruning Agent User's Guide

Related publications

The publications in related information centers provide useful information.

See the following information centers, which you can find by accessing Tivoli Documentation Central (http://www.ibm.com/tivoli/documentation):

- · IBM Tivoli Monitoring
- IBM Tivoli Netcool/OMNIbus
- · IBM Tivoli Application Dependency Discovery Manager
- IBM Tivoli Enterprise Console

Other sources of documentation

You can obtain additional technical documentation about monitoring products from other sources.

See the following sources of technical documentation about monitoring products:

Service Management Connect (SMC)

See the introductory information about SMC (http://www.ibm.com/developerworks/servicemanagement/).

For information about Tivoli products, see the Application Performance Management community on SMC (http://www.ibm.com/developerworks/servicemanagement/apm/index.html).

Connect, learn, and share with Service Management professionals. Get access to developers and product support technical experts who provide their perspectives and expertise. You can use SMC for these purposes:

- Become involved with transparent development, an ongoing, open engagement between external
 users and developers of Tivoli products where you can access early designs, sprint demos, product
 roadmaps, and pre-release code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and Integrated Service Management.
- Benefit from the expertise and experience of others using blogs.
- Collaborate with the broader user community using wikis and forums.
- IBM Integrated Service Management Library (http://www.ibm.com/software/brandcatalog/ismlibrary/) is an online catalog that contains integration documentation as well as other downloadable product extensions.
- IBM Redbook publications (http://www.redbooks.ibm.com/) include Redbooks[®] publications, Redpapers, and Redbooks technotes that provide information about products from platform and solution perspectives.

- Technotes (http://www.ibm.com/support/entry/portal/software), which are found through the IBM Software Support website, provide the latest information about known product limitations and workarounds.
- · Tivoli wikis

Tivoli Wiki Central (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 (http://www-10.lotus.com/ldd/ tivmonitorwiki.nsf) provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Manager products.
- Tivoli System z[®] Monitoring and Application Management Wiki (http://www.ibm.com/ developerworks/wikis/display/tivoliomegamon/Home) provides information about the OMEGAMON XE products, Tivoli NetView® for z/OS®, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.

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. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, 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:

© IBM 2009. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2009. 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 (www.ibm.com/legal/copytrade.shtml).

Intel, Intel logo, and Intel Xeon, are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



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

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

Microsoft and Windows 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.

Index

A	attributes (continued)			
activities 97	Last Collection Duration 79, 84			
	Last Collection Finished 79, 84			
additional information	Last Collection Start 79, 84			
attributes 57	Last Matched Time 62			
situations 91	Last Modification Time 66			
Take Action commands 95	Last Unmatched Time 63			
Workspaces 51	LFAProfiles 58			
agent	Log File RegEx Statistics 60			
functions 1	Log File Status 63			
problems and workarounds 115	LogfileEvents 67			
Agent Management Services 3	LogfileProfileEvents 71			
All Regular Expression statistics workspace 53	Logname 68, 72			
Attrib Name attribute 60	Max Processor Time 61			
attribute group 58	Min Processor Time 62			
attribute groups	msg 68, 72			
LFAProfiles 58	Node 58, 60, 63, 67, 71, 76, 81, 86			
list of all 57	Num Records Matched 65			
Log File RegEx Statistics 60	Num Records Not Matched 65			
Log File Status 63	Num Records Processed 66			
LogfileEvents 67	Number of Collections 80, 85			
LogfileProfileEvents 71	Object Name 76, 81			
overview 57	Object Status 77, 82			
Performance Object Status 76	Object Type 77, 81			
pro Performance Object Status 81	Occurrence Count 70, 74			
Thread Pool Status 86	overview 57			
attributes 58	Performance Object Status 76			
additional information 57	pro Performance Object Status 81			
Attrib Name 60	Query Name 76, 81			
Average Collection Duration 80, 84	Refresh Interval 80, 85			
Average Processor Time 61	RegEx Pattern 62, 64			
Cache Hit Percent 80, 85	Remote Host 67			
Cache Hits 80, 85	RemoteHost 71, 76			
Cache Misses 80, 85	Result Type 63			
Class 67, 72	Subnode Affinity 59			
Codepage 66	Subnode Config File 59			
Current File Position 66	Subnode Description 59			
Current File Size 66	Subnode MSN 59			
CustomInteger 70, 75	Subnode Resource Name 59			
CustomInteger2 70, 75	Subnode Type 59			
CustomInteger3 71, 75	Subnode Version 59			
CustomSlot1 68, 72	Table Name 60, 64			
CustomSlot10 69, 74	Thread Pool Active Threads 87			
CustomSlot2 68, 73	Thread Pool Avg Active Threads 87			
CustomSlot3 68, 73	Thread Pool Avg Job Wait 89			
CustomSlot4 68, 73	Thread Pool Avg Queue Length 88			
CustomSlot5 69, 73	Thread Pool Max Active Threads 87			
CustomSlot6 69, 73	Thread Pool Max Queue Length 88			
CustomSlot7 69, 73	Thread Pool Max Size 86			
CustomSlot8 69, 74	Thread Pool Min Active Threads 87			
CustomSlot9 69, 74	Thread Pool Min Queue Length 88			
EIFEvent 68, 72	Thread Pool Queue Length 88			
Error Code 77, 82	Thread Pool Size 86			
Event Type 70, 74	Thread Pool Status 86			
File Name 64	Thread Pool Total Jobs 89			
File Status 64	Timestamp 59, 60, 64, 67, 72, 76, 81, 86			
File Type 64	Total Processor Time 61			
Filter Count 62	Average Collection Duration attribute 80, 84			
Filter Count Matched 62	Average Processor Time attribute 61			
Filter Count Unmatched 62				
Filter Number 61				
Intervals Skipped 80, 85				

C	Error Code attribute 77, 82		
Cache Hit Percent attribute 80, 85	event		
Cache Hits attribute 80, 85	mapping 131 Event filtering and summarization 43		
Cache Misses attribute 80, 85	summarization attributes 45		
calculate historical data disk space 89			
capacity planning for historical data 89	summary interval 44		
Class attribute 67, 72	viewing in the Tivoli Enterprise Portal 45		
Codepage attribute 66	Event Type attribute 70, 74		
commands			
Take Action 95	F		
components 2	Г		
IBM Tivoli Monitoring 2	File Name attribute 64		
configuration	File Status attribute 64		
after installation 6	File Type attribute 64		
centralized configuration 10	Filter Count attribute 62		
command line 8	Filter Count Matched attribute 62		
configuration file 23	Filter Count Unmatched attribute 62		
configuration files 9	Filter Number attribute 61		
configuration panel 6	Filtering events 44		
configuring the monitoring agent 6	Format file 33		
problems and workarounds 111	changing 10		
remote installation and configuration 14	Format file specifications 33		
configuration file 23	inheritance 37		
Configuration file	keywords 40		
changing 10	mappings 38		
configuration files 9	value specifiers 39		
Configuring the ObjectServer 16	Maximum message length 41		
CPU	Multi-line 38		
CPU throttling 10	Forwarding events		
CPU throttling 123	to Tivoli Netcool OMNIbus ObjectServers 15, 16		
Current File Position attribute 66			
Current File Size attribute 66			
CustomInteger1 attribute 70, 75	Н		
CustomInteger2 attribute 70, 75	High Average CPU Regular Expressions workspace 53		
CustomInteger3 attribute 71, 75	High Total CPU Regular Expressions workspace 53		
CustomSlot1 attribute 68, 72	historical data		
CustomSlot10 attribute 69, 74	calculate disk space 89		
CustomSlot2 attribute 68, 73	disk capacity planning 89		
CustomSlot3 attribute 68, 73	1 71 0		
CustomSlot4 attribute 68, 73			
CustomSlot5 attribute 69, 73	I		
CustomSlot6 attribute 69, 73	IDA (III) I I A (III) A		
CustomSlot7 attribute 69, 73	IBM Tivoli Monitoring 2		
CustomSlot8 attribute 69, 74	overview 1		
CustomSlot9 attribute 69, 74	IBM Tivoli Monitoring version 5		
	installation		
_	commands		
D	tacmd addSystem 14		
data collection 4	problems and workarounds 111		
Data Collection Status (remote) workspace 53	Remote installation and configuration 14		
Data Collection Status workspace 53	tacmd addSystem command 14		
data sources 4, 6	Integrated Service Management Library documentation 140		
descriptions 92	interface		
detailed 105	user 3		
developerWorks website 140	Intervals Skipped attribute 80, 85		
disk capacity planning for historical data 89			
disk space 5	1/		
documentation	K		
See publications	KLO_Log_Agent_Config_Error situation 92		
duplicate events	KLO_PRO_Log_Agent_Config_Error situation 93		
detecting and filtering 43	_ 0_ 0 _ 0 _		
detecting and intering 10			
	L		
E	Last Callastian Donation attails to 70 04		
	Last Collection Duration attribute 79, 84		
EIFEvent attribute 68, 72	Last Collection Finished attribute 79, 84		
enhancements 1	Last Collection Start attribute 79, 84		

Last Matched Time attribute 62 Last Modification Time attribute 66 Last Unmatched Time attribute 63 LFAProfiles workspaces descriptions 55 LFAProfiles attribute group 58 LFAProfiles workspace 55 Linux compatibility libraries 5 Log File agent performance considerations 127 Log File RegEx Statistics attribute group 60 Log File Status attribute group 63 Log Files workspaces descriptions 52 Log Files workspace 52 Logfile Events (remote) workspace 54 Logfile Events (v6) workspace 54 LogfileEvents workspaces descriptions 55 LogfileEvents attribute group 67 LogfileEvents workspace 55 LogfileEvents workspace 55 LogfileProfileEvents attribute group 71 Logname attribute 68, 72	Probe for Tivoli EIF (continued) running on Unix 18 running on windows 18 problems and workarounds 111 agent-specific 115 agent-specific workspaces 125 configuration 111 install 111 remote deployment 115 situations 127 Take Action commands 130 workspaces 125 publications 139, 140 developerWorks website 140 IBM Tivoli Monitoring 139 Integrated Service Management Library 140 prerequisite 139 Redbooks 140 related 140 Technotes 140 wikis 140 Q queries, using attributes 57			
Zogranic utilibute 00,72	Query Name attribute 76, 81			
M	R			
Max Processor Time attribute 61 Min Processor Time attribute 62 monitoring profiles 18 msg attribute 68, 72	ras1 107 Redbooks 140 Refresh Interval attribute 80, 85 RegEx Pattern attribute 62, 64 remote			
N	installation and configuration 14			
new in this release 1 Node attribute 58, 60, 63, 67, 71, 76, 81, 86 Num Records Matched attribute 65 Num Records Not Matched attribute 65 Num Records Processed attribute 66 Number of Collections attribute 80, 85	remote deployment problems and workarounds 115 Remote Host attribute 67 remote log file monitoring troubleshooting 124 Remote log file monitoring 11 Encryption Pass phrase 13			
0	Password 13 RemoteHost attribute 71, 76			
Object Name attribute 76, 81 Object Status attribute 77, 82 Object Type attribute 77, 81 Occurrence Count attribute 70, 74 overview IBM Tivoli Monitoring 1	requirements configuration 5 installation 5 software product compatibility report 6 Requirements for the monitoring agent 5 Result Type attribute 63			
P	S			
performance considerations 127 Performance Object Status situations 92 Performance Object Status attribute group 76 policies 97 prerequisite publications 139 pro Performance Object Status situations 93 pro Performance Object Status attribute group 81 Probe for Tivoli EIF 17 properties 18	situations 92 additional information predefined, defined 91 KLO_Log_Agent_Config_Error 92 KLO_PRO_Log_Agent_Config_Error 93 overview 91 predefined 91 problems and workarounds 127 Situation Editor 91 situations, using attributes 57 software product compatibility report 6 Starting the agent 15			

Subnode Affinity attribute 59 Subnode Config File attribute 59 Subnode Description attribute 59 Subnode MSN attribute 59 Subnode Resource Name attribute 59 Subnode Type attribute 59 Subnode Version attribute 59 subnodes how they work 18 log file agent subnodes 18 monitoring profiles 18	views All Regular Expression statistics workspace 53 Data Collection Status (remote) workspace 53 Data Collection Status workspace 53 High Average CPU Regular Expressions workspace 53 High Total CPU Regular Expressions workspace 53 LFAProfiles workspace 55 Log Files workspace 52 Logfile Events (remote) workspace 54 Logfile Events (v6) workspace 54, 55 LogfileEvents workspace 55			
T	Tivoli Log File Profile (remote) workspace 54			
Table Name attribute 60, 64 Take Action commands additional information 95 overview 95 predefined 95, 97 problems and workarounds 130 Technotes 140 Thread Pool Active Threads attribute 87 Thread Pool Avg Active Threads attribute 87 Thread Pool Avg Job Wait attribute 89 Thread Pool Avg Queue Length attribute 88 Thread Pool Max Queue Length attribute 87 Thread Pool Max Size attribute 86 Thread Pool Min Active Threads attribute 87 Thread Pool Min Active Threads attribute 87 Thread Pool Min Queue Length attribute 87 Thread Pool Min Queue Length attribute 88 Thread Pool Size attribute 86 Thread Pool Size attribute 86 Thread Pool Size attribute 87 Thread Pool Size attribute 89 Timestamp attribute 59, 60, 64, 67, 72, 76, 81, 86 Tivoli Enterprise Console event mapping 131 Tivoli Log File Profile (remote) workspace 54 Tivoli Netcool/OMNIbus version 5 Total Processor Time attribute 61 trace turn off 110 turn on 109 trace settings 107 tracing 105 troubleshooting 99, 123, 124 agent-specific 115 agent-specific workspaces 125 installation 111 problems and workarounds 111 remote deployment 115 situations 127 Take Action commands 130 turn off trace 110 turn on trace 109 uninstallation 111	Wwikis 140 Windows 2008 event log 137 workarounds 111 Workflow Editor 97 workspaces All Regular Expression statistics 53 Data Collection Status 53 Data Collection Status 53 Data Collection Status (remote) 53 descriptions 52 High Average CPU Regular Expressions 53 LFAProfiles 55 Log Files 52 Log file Events (remote) 54 Logfile Events (v6) 54, 55 LogfileEvents 55 predefined 52 problems and workarounds 125 Tivoli Log File Profile (remote) 54 Unmatched Regular Expressions 55 Workspaces additional information 51 overview 51			
workspaces 125				

Unmatched Regular Expressions workspace 55 user interface options 3

IBM.

Printed in USA

SC14-7484-04

